9.4

IBM MQ 配置参考

注

在使用本资料及其支持的产品之前，请阅读第 225 页的『声明』中的信息。

本版本适用于 IBM® MQ V 9 发行版 4 以及所有后续发行版和修订版，直到在新版本中另有声明为止。

当您向 IBM 发送信息时，授予 IBM 以它认为适当的任何方式使用或分发信息的非独占权利，而无需对您承担任何责任。

© Copyright International Business Machines Corporation 2007, 2025.

内容

配置参考 5
示例: 为 IBM MQ 设置跨平台通信 5
如何使用跨平台通信示例 7
示例: 在 AIX 上设置 IBM MQ 跨平台通信 8
示例: 在 IBM i 上设置 IBM MQ 跨平台通信 14
示例: 在 Linux 上设置 IBM MQ 跨平台通信 29
示例: 在 Windows 上设置 IBM MQ 跨平台通信 35
示例: 在 z/OS 上设置 IBM MQ 跨平台通信 41
Example: setting up IBM MQ cross-platform communication on z/OS using QSGs 45
Example: setting up IBM MQ cross-platform communication for intra-group queuing on z/OS 52
IBM MQ 文件系统许可权已应用于 /var/mqm 59
/opt/mqm 中具有 mqm 的 setuid 的 IBM MQ 文件许可权 62
Windows 上的 IBM MQ 文件系统许可权 63
队列的命名限制 64
其他对象的命名限制 66
队列名称解析 67
什么是队列名称解析? 69
如何解析别名，远程队列和集群队列的目标对象属性? 69
系统和缺省对象 69
SYSTEM.BASE.TOPIC 73
配置文件节信息 74
用于分布式排队的配置文件节 77
通道属性 78
MQSC 关键字 (A-B) 的通道属性 82
MQSC 关键字的通道属性 (C) 85
MQSC 关键字 (D-L) 的通道属性 91
MQSC 关键字的通道属性 (M) 97
MQSC 关键字的通道属性 (N-R) 101
MQSC 关键字的通道属性 (S) 105
MQSC 关键字的通道属性 (T-Z) 109
IBM MQ 集群命令和属性 111
通道定义命令上可用的集群属性 112
队列定义命令上可用的集群属性 114
队列管理器定义命令上可用的集群属性 116
DISPLAY CLUSQMGR 117
刷新集群 118
重置集群: 强制从集群中移除队列管理器 119
SUSPEND QMGR, RESUME QMGR 和集群 121
集群中的工作负载均衡 121
Asynchronous behavior of CLUSTER commands on z/OS 128
通道程序 129
IBM i 上的相互通信作业 129
IBM i 上的通道状态 130
示例: 在 AIX, Linux, and Windows 上规划消息通道 130
设置 AIX, Linux, and Windows 的消息通道示例 131
运行和展开 AIX, Linux, and Windows 的示例 133
示例: 在 IBM i 上规划消息通道 133
在 IBM i 上设置消息通道代理程序 135
运行和展开 IBM i 的示例 137
示例: 在 z/OS 上规划消息通道 138
在 z/OS 上设置消息通道代理程序 139
Running and expanding the example for z/OS 141

Example: planning a message channel for z/OS using queue sharing groups....141
设置队列共享组定义和不在队列共享组中的队列管理器 QM3....143
运行 z/OS 的队列共享组示例....144
使用别名来引用 MQ 库....144
Managed File Transfer 配置参考信息....145
在 MFT 属性中使用环境变量....145
MFT installation.properties 文件....146
MFT `agent.properties` 文件....149
MFT coordination.properties 文件....167
MFT command.properties 文件....170
MFT logger.properties 文件....173
由 LogTransfer 函数生成的输出....181
MFT 的 Java 系统属性....184
MFT 的 SHA-2 CipherSpec 和 CipherSuite....185
MFT 文件记录器配置文件....185
SSQFCMD 库....192
SYSTEM.FTE 主题....194
MFT 代理队列设置....195
MFT 系统队列和系统主题....196
MFT 对象命名约定....197
MFT 代理状态消息....198
IBM MQ Internet Pass-Thru 配置参考信息....199
MQIPT 属性的摘要....200
MQIPT 全局属性....205
MQIPT 路由属性....208
`mqiptAdmin` 属性....223

声明....225
编程接口信息....226
商标....226

配置参考

使用此部分中的参考信息来帮助您配置 IBM MQ。

以下子主题中提供了配置参考信息:

相关任务

配置

z/OS 配置 z/OS

示例: 为 IBM MQ 设置跨平台通信

此示例显示如何通过配置 IBM MQ 发送方和接收方通道来建立工作的 IBM MQ 网络，以在所有受支持协议上的平台之间启用双向消息流。

开始之前

配置示例假定特定网络基础结构适用于特定平台:

- z/OS z/OS 使用 3745 网络控制器 (或等效) 进行通信

还假定对于 SNA， VTAM 和网络控制程序 (NCP) 中的所有必需定义都已就绪，并且已激活，以便 LAN 连接的平台通过广域网 (WAN) 进行通信。 同样，对于 TCP，假定名称服务器功能可用，可以使用域名服务器，也可以使用本地保存的表 (例如，主机文件)。

示例配置涵盖以下网络软件产品:

- SNA
  - Windows IBM Windows 的 Personal Communications
  - AIX IBM 通信服务器 AIX
  - IBM i IBM i
  - OS/390
- TCP
  - Windows Microsoft Windows
  - AIX AIX
  - IBM i IBM i
  - z/OS z/OS 的 TCP
- Windows NetBIOS
- Windows SPX

有关受支持的通信协议和软件的更多信息，请参阅 系统要求 IBM MQ。

关于此任务

此示例使用发送方和接收方通道。 要使用不同于发送方/接收方的通道类型，请 DEFINE CHANNEL (定义新通道)。

第 6 页的图 1 是单个通道以及与其关联的 IBM MQ 对象的概念表示。

© Copyright IBM Corp. 2007, 2025

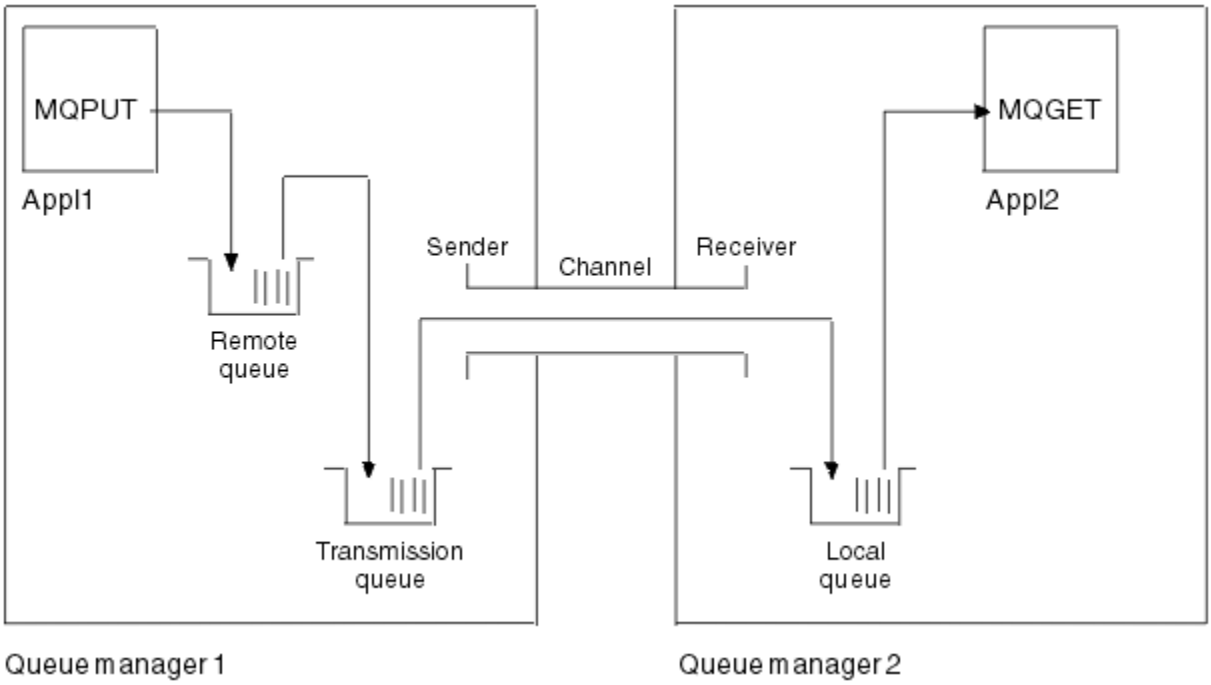

图 *1:* 要在示例配置中设置的 *IBM MQ* 通道

此示例是一个简单示例，旨在仅引入 IBM MQ 网络的基本元素。 它不会演示如何使用 触发通道中描述的触发。

此网络中的对象为:

- 远程队列
- 传输队列
- 本地队列
- 发送方通道
- 接收方通道

Appl1 和 Appl2 都是应用程序; Appl1 正在放置消息， Appl2 正在接收消息。

Appl1 将消息放入远程队列。 此远程队列的定义指定目标队列管理器，该队列管理器上的本地队列以及此本地队列管理器上的传输队列的名称。

当队列管理器从 Appl1 接收到将消息放入远程队列的请求时，队列管理器根据队列定义确定目标是远程的。 因此，它将消息与传输头直接放在定义中指定的传输队列上。 消息将保留在传输队列上，直到通道变为可用为止，这可能会立即发生。

发送方通道在其定义中具有对一个且仅一个传输队列的引用。 当通道启动时，在其正常运行期间的其他时间，它会查看此传输队列并将其上的任何消息发送到目标系统。 消息在其传输头中包含目标队列和队列管理器的详细信息。

相互通信示例详细描述了针对各种平台组合创建上述每个对象的过程。

在目标队列管理器上，本地队列和通道接收方需要定义。 这些对象相互独立操作，因此可以按任何顺序创建。

在本地队列管理器上，远程队列，传输队列和通道的发送方端都需要定义。 由于远程队列定义和通道定义都引用了传输队列名称，因此建议先创建传输队列。

过程

1. 阅读 第 7 页的『如何使用跨平台通信示例』中的信息。
2. 遵循相应平台的指示信息以建立网络连接并定义通道。

a) AIX
请参阅第 8 页的『示例: 在 AIX 上设置 IBM MQ 跨平台通信』

b) IBM i
请参阅第 14 页的『示例: 在 IBM i 上设置 IBM MQ 跨平台通信』

c) Linux
请参阅第 29 页的『示例: 在 Linux 上设置 IBM MQ 跨平台通信』

d) Windows
请参阅第 35 页的『示例: 在 Windows 上设置 IBM MQ 跨平台通信』

e) z/OS
请参阅第 41 页的『示例: 在 z/OS 上设置 IBM MQ 跨平台通信』

f) z/OS
请参阅第 45 页的『Example: setting up IBM MQ cross-platform communication on z/OS using QSGs』

g) z/OS
请参阅第 52 页的『Example: setting up IBM MQ cross-platform communication for intra-group queuing on z/OS』

相关任务

配置分布式队列

在 z/OS 上设置与其他队列管理器的通信

如何使用跨平台通信示例

用于为 IBM MQ 设置跨平台通信的示例配置描述了在单个平台上执行的用于设置与另一个平台的通信的任务。 然后，这些示例描述了用于建立到该平台的工作通道的任务。

在可能的情况下，目的是使信息尽可能通用。 因此，要在不同平台上连接任何两个队列管理器，只需要引用相关的两个部分。 因此，将突出显示任何偏差或特殊情况。 您还可以连接在同一平台上 (在不同机器上或同一机器上) 运行的两个队列管理器。 在这种情况下，所有的信息都可以从一个部分中得到。

ALW 在 AIX, Linux, and Windows 上，开始遵循针对平台的指示信息之前，必须设置各种环境变量。 通过输入下列其中一个命令来执行此操作:

- Linux AIX 在 AIX and Linux 上:

  ```
  MQ_INSTALLATION_PATH/bin/setmqenv
  ```

  其中 *MQ_INSTALLATION_PATH* 是指 IBM MQ 的安装位置。 此命令为您当前正在使用的 shell 设置环境变量。 如果打开另一个 shell，那么必须再次输入该命令。

- Windows 在 Windows 上:

  ```
  MQ_INSTALLATION_PATH/bin/setmqenv
  ```

  其中 *MQ_INSTALLATION_PATH* 是指 IBM MQ 的安装位置。

在一些示例中，您可以找到样本配置中使用的参数。 提供了每个参数的简短描述以及有关在系统中查找等效值的位置的一些指导信息。 当您有一组自己的值时，请确保在处理本节中的示例时使用这些值。

这些示例不涵盖如何在使用集群的情况下设置通信。 有关在使用集群时设置通信的信息，请参阅 配置队列管理器集群。 此处提供的通信配置值仍然适用。

以下平台有一些示例配置:

- AIX 第 8 页的『示例: 在 AIX 上设置 IBM MQ 跨平台通信』

- IBM i 第 14 页的『示例: 在 IBM i 上设置 IBM MQ 跨平台通信』
- Linux 第 29 页的『示例: 在 Linux 上设置 IBM MQ 跨平台通信』
- Windows 第 35 页的『示例: 在 Windows 上设置 IBM MQ 跨平台通信』
- z/OS 第 41 页的『示例: 在 z/OS 上设置 IBM MQ 跨平台通信』
- z/OS 第 45 页的『Example: setting up IBM MQ cross-platform communication on z/OS using QSGs』
- z/OS 第 52 页的『Example: setting up IBM MQ cross-platform communication for intra-group queuing on z/OS』

IT 职责

要了解示例中使用的术语，请将以下准则作为起点。

- 系统管理员: 为特定平台安装和配置软件的人员 (或人员组)。
- 网络管理员: 控制 LAN 连接， LAN 地址分配，网络命名约定和其他网络任务的人员。 此人员可以在单独的组中，也可以是系统管理组的一部分。

  在大多数 z/OS 安装中，有一个组负责更新 ACF/VTAM， ACF/NCP 和 TCP/IP 软件以支持网络配置。 此组中的人员是将任何 IBM MQ 平台连接到 IBM MQ for z/OS 时所需的主要信息源。 它们还会影响或要求 LAN 上的网络命名约定，您必须先验证其控制范围，然后再创建定义。
- 在我们可以更清楚地描述人员职责的情况下，会指示特定类型的管理员 (例如 CICS 管理员)。

示例配置部分不会尝试指示谁负责并能够设置每个参数。 一般来说，可能涉及到几个不同的人。

相关任务

第 5 页的『示例: 为 IBM MQ 设置跨平台通信』
此示例显示如何通过配置 IBM MQ 发送方和接收方通道来建立工作的 IBM MQ 网络，以在所有受支持协议上的平台之间启用双向消息流。

相关参考

setmqenv

示例: 在 AIX 上设置 IBM MQ 跨平台通信

此示例显示如何在另一个平台上设置从 IBM MQ on AIX 到 IBM MQ 的通信链路，并建立到该平台的工作通道。

开始之前

有关此示例以及如何使用此示例的背景信息，请参阅 第 5 页的『示例: 为 IBM MQ 设置跨平台通信』 和 第 7 页的『如何使用跨平台通信示例』。

关于此任务

此示例涵盖在 AIX 上设置从 IBM MQ 到以下平台的跨平台通信:

- Windows Windows
- Linux Linux
- IBM i IBM i
- z/OS z/OS
- VSE/ESA

过程

1. 使用下列其中一个选项建立网络连接。
   - 建立 LU 6.2 连接。 有关配置基于 TCP/IP 的 SNA 的更多信息，请参阅 Communications Server for AIX Library。
   - 建立 TCP 连接。

     必须先显式启动侦听器，然后才能启动任何通道。 它允许接收通道自动启动，以响应来自入站发送通道的请求。 使用以下命令来启动 IBM MQ for TCP 侦听器:

     ```
     runmqlsr -t tcp
     ```

     a. 编辑文件 /etc/services。

        注: 要编辑 /etc/services 文件，您必须以超级用户或 root 用户身份登录。 如果该文件中没有以下行，请按如下所示添加该文件:

        ```
        MQSeries       1414/tcp      # MQSeries channel listener
        ```

     b. 编辑文件 /etc/inetd.conf。 如果该文件中没有以下行，请按如下所示添加该文件，将 *MQ_INSTALLATION_PATH* 替换为安装了 IBM MQ 的高级目录:

        ```
        MQSeries stream tcp nowait root MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta
        [-m queue.manager.name]
        ```

     c. 输入命令 refresh -s inetd。

     注: 必须将 **root** 添加到 mqm 组。 您无需将主组设置为 mqm。 只要 mqm 在组集中，您就可以使用这些命令。 如果仅运行使用队列管理器的应用程序，那么您不需要 mqm 组权限。
2. 建立连接后，定义一些通道，如 第 9 页的『在 AIX 上配置通道』中所述。

AIX 在 AIX 上配置通道

要为 AIX 上的示例配置配置 IBM MQ ，请完成队列管理器的基本配置步骤，然后配置发送方和接收方通道。

开始之前

注:

1. 在开始安装过程之前，请确保首先创建了 mqm 用户和组，并设置密码。
2. 如果由于文件系统中的空间不足而导致安装失败，那么可以使用命令 smit C sna 来增大大小，如下所示。 (使用 df 来显示文件系统的状态。 这指示已满的逻辑卷。)

   ```
   -- Physical and Logical Storage
     -- File Systems
       -- Add / Change / Show / Delete File Systems
         -- Journaled File Systems
           -- Change/Show Characteristics of a Journaled File System
   ```

3. 使用以下命令启动任何通道:

   ```
   runmqchl -c channel.name
   ```

4. 样本程序安装在 *MQ_INSTALLATION_PATH*/samp 中，其中 *MQ_INSTALLATION_PATH* 表示安装 IBM MQ 的高级目录。
5. 错误日志存储在 /var/mqm/qmgrs/ *qmgrname* /errors 中。
6. 在 AIX 上，可以使用标准 IBM MQ 跟踪命令或使用 AIX 系统跟踪来启动 IBM MQ 组件的跟踪。 有关 IBM MQ 跟踪和 AIX 系统跟踪的更多信息，请参阅 使用跟踪 。
7. 当您使用命令解释器 **runmqsc** 输入管理命令时，行尾的 + 表示下一行是连续行。 确保最后一个参数与连续字符之间有空格。

过程

1. 要执行队列管理器的基本配置，请完成以下步骤:
   a) 使用以下命令从 AIX 命令行创建队列管理器:

```
crtmqm -u dlqname -q aix
```

   其中:

   aix
   是队列管理器的名称

   -q
   指示这将成为缺省队列管理器

   -u *dlqname*
   指定无法传递的消息队列的名称

   此命令创建队列管理器和一组缺省对象。

   b) 使用以下命令从 AIX 命令行启动队列管理器:

```
strmqm aix
```

   其中， *aix* 是创建队列管理器时提供给该队列管理器的名称。

   c) 从 AIX 命令行启动 **runmqsc** ，并通过输入以下命令使用它来创建无法传递的消息队列:

```
def ql (dlqname)
```

   其中， *dlqname* 是创建队列管理器时提供给无法传递的消息队列的名称。

2. 配置示例配置的通道。

   有关以下示例中使用的参数的更多信息，请参阅 第 11 页的『AIX 的通道配置参数』。 在每种情况下，示例都显示 MQSC 命令。 从 AIX 命令行启动 **runmqsc** 并依次输入每个命令，或者将这些命令构建到命令文件中。

   Windows 这些示例用于将 AIX 上的 IBM MQ 与 Windows 上的 IBM MQ 连接。 要在另一个平台上连接到 IBM MQ ，请使用 第 11 页的『AIX 的通道配置参数』 中表中的相应值，而不是 Windows 的值。

   a) 定义发送方通道，如以下示例中所示:

   - 使用 SNA:

```
def ql (WINNT) +                                F
    usage(xmitq) +
    replace

def qr (WINNT.REMOTEQ) +                        D
    rname(WINNT.LOCALQ) +                       E
    rqmname(WINNT) +                            C
    xmitq(WINNT) +                              F
    replace

def chl (AIX.WINNT.SNA) chltype(sdr) +          G
    trptype(lu62) +
    conname('WINNTCPIC') +                      17
    xmitq(WINNT) +                              F
    replace
```

   - 使用 TCP:

```
def ql (WINNT) +                                F
    usage(xmitq) +
    replace

def qr (WINNT.REMOTEQ) +                        D
    rname(WINNT.LOCALQ) +                       E
    rqmname(WINNT) +                            C
    xmitq(WINNT) +                              F
    replace

def chl (AIX.WINNT.TCP) chltype(sdr) +          H
```

```
    trptype(tcp) +
    conname(remote_tcpip_hostname) +
    xmitq(WINNT) +                              F
    replace
```

b) 定义接收方通道，如以下示例中所示:

- 使用 SNA:

```
def ql (AIX.LOCALQ) replace                   B

def chl (WINNT.AIX.SNA) chltype(rcvr) +       I
    trptype(lu62) +
    replace
```

- 使用 TCP:

```
def ql (WINNT) +                              F
    usage(xmitq) +
    replace

def qr (WINNT.REMOTEQ) +                      D
    rname(WINNT.LOCALQ) +                     E
    rqmname(WINNT) +                          C
    xmitq(WINNT) +                            F
    replace

def chl (AIX.WINNT.TCP) chltype(sdr) +        H
    trptype(tcp) +
    conname(remote_tcpip_hostname) +
    xmitq(WINNT) +                            F
    replace
```

注: 有其他方法可确保在发送方通道启动对话时正确激活 SNA 接收方通道。

在 AIX Communications Server 配置过程中，创建了一个 LU 6.2 TPN 概要文件，其中包含 TP 可执行程序的完整路径。 在此示例中，该文件名为 u/interops/AIX.crs6a。 您可以选择名称，但请考虑在其中包含队列管理器的名称。 可执行文件的内容必须是:

```
#!/bin/sh
MQ_INSTALLATION_PATH/bin/amqcrs6a -m aix
```

其中， *aix* 是队列管理器名称 (A) ， *MQ_INSTALLATION_PATH* 是安装 IBM MQ 的高级目录。 创建此文件后，通过运行以下命令将其启用以执行:

```
chmod 755 /u/interops/AIX.crs6a
```

作为创建可执行文件的替代方法，您可以使用命令行参数在 "添加 LU 6.2 TPN 概要文件" 面板上指定路径。

在这两种方法之一中指定路径可确保当发送方通道启动对话时正确激活 SNA 接收方通道。

AIX *AIX* 的通道配置参数

为 AIX 上的示例配置配置通道所需的参数。

步骤 第 10 页的『2』 (共 第 9 页的『在 AIX 上配置通道』 步) 描述了要在 AIX 队列管理器上执行的配置，以实现 第 5 页的『示例: 为 IBM MQ 设置跨平台通信』 中描述的通道。 第 9 页的『在 AIX 上配置通道』 中的示例用于连接 IBM MQ for IBM i 和 IBM MQ for Windows。 要在另一个平台上连接到 IBM MQ ，请使用相应表中的值来代替 Windows 的值。

注: 粗体 中的词是建议值，并反映在这些示例中使用的 IBM MQ 对象的名称。 您可以在产品安装中更改这些值，但如果更改了这些值，请确保在使用本部分中的示例时使用自己的值

本地节点的定义

表 *1:* 本地节点定义的配置示例			
标识	**参数名**	**参考**	**使用的示例**
A	队列管理器名称		**AIX**
B	本地队列名称		**AIX.LOCALQ**

连接到 Windows 上的 IBM MQ

Windows

表的此部分中的值必须与 第 39 页的『Windows 的通道配置参数』中使用的值匹配，如所指示。

表 *2:* 用于连接到 *Windows* 上的 *IBM MQ* 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**winnt**
D	远程队列名称		**WINNT.REMOTEQ**
E	远程系统上的队列名称	B	**WINNT.LOCALQ**
F	传输队列的名称		**winnt**
G	发送方 (SNA) 通道名称		**AIX.WINNT.SNA**
H	发送方 (TCP/IP) 通道名称		**AIX.WINNT.TCP**
I	接收方 (SNA) 通道名称	G	**WINNT.AIX.SNA**
J	接收方 (TCP) 通道名称	H	**WINNT.AIX.TCP**

连接到 Linux 上的 IBM MQ

Linux

表的此部分中的值必须与 第 33 页的『Linux 的通道配置参数』中使用的值匹配，如所指示。

表 *3:* 用于连接到 *Linux* 上的 *IBM MQ* 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**Linux**
D	远程队列名称		**LINUX.REMOTEQ**
E	远程系统上的队列名称	B	**LINUX.LOCALQ**
F	传输队列的名称		**Linux**
G	发送方 (SNA) 通道名称		**AIX.LINUX.SNA**
H	发送方 (TCP/IP) 通道名称		**AIX.LINUX.TCP**
I	接收方 (SNA) 通道名称	G	**LINUX.AIX.SNA**
J	接收方 (TCP/IP) 通道名称	H	**LINUXAIX。TCP**

连接到 IBM i 上的 IBM MQ

IBM i

表的此部分中的值必须与 第 27 页的『IBM i 的通道配置参数』中使用的值匹配，如所指示。

表 4: 用于连接到 *IBM i* 上的 *IBM MQ* 的配置示例			
标识	**参数名称**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**AS400**
D	远程队列名称		**AS400.REMOTEQ**
E	远程系统上的队列名称	B	**AS400.LOCALQ**
F	传输队列的名称		**AS400**
G	发送方 (SNA) 通道名称		**AIX.AS400.SNA**
H	发送方 (TCP) 通道名称		**AIX.AS400.TCP**
I	接收方 (SNA) 通道名称	G	**AS400.AIX.SNA**
J	接收方 (TCP) 通道名称	H	**AS400.AIX。TCP**

连接到 IBM MQ for z/OS

z/OS

表的此部分中的值必须与 第 43 页的『z/OS 的通道配置参数』中使用的值匹配，如所指示。

表 5: 用于连接到 *IBM MQ for z/OS* 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**MVS**
D	远程队列名称		**MVS.REMOTEQ**
E	远程系统上的队列名称	B	**MVS.LOCALQ**
F	传输队列的名称		**MVS**
G	发送方 (SNA) 通道名称		**AIX.MVS.SNA**
H	发送方 (TCP) 通道名称		**AIX.MVS.TCP**
I	接收方 (SNA) 通道名称	G	**MVS.AIX.SNA**
J	接收方 (TCP) 通道名称	H	**MVS。AIX。TCP**

使用队列共享组连接到 IBM MQ for z/OS

z/OS

表的此部分中的值必须与 第 51 页的『Shared channel configuration parameters』中使用的值匹配，如所指示。

表 6: 使用队列共享组连接到 *IBM MQ for z/OS* 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**QSG**
D	远程队列名称		**QSG.REMOTEQ**
E	远程系统上的队列名称	B	**QSG.SHAREDQ**
F	传输队列的名称		**QSG**
G	发送方 (SNA) 通道名称		**AIX.QSG.SNA**
H	发送方 (TCP) 通道名称		**AIX.QSG.TCP**

表 6: 使用队列共享组连接到 *IBM MQ for z/OS* 的配置示例 (继续)			
标识	**参数名**	**参考**	**使用的示例**
I	接收方 (SNA) 通道名称	G	**QSG.AIX.SNA**
J	接收方 (TCP) 通道名称	H	**QSG.AIX。TCP**

IBM i 示例: 在 IBM i 上设置 IBM MQ 跨平台通信

此示例显示如何在另一个平台上设置从 IBM MQ on IBM i 到 IBM MQ 的通信链路，并建立到该平台的工作通道。

开始之前

有关此示例以及如何使用此示例的背景信息，请参阅 第 5 页的『示例: 为 IBM MQ 设置跨平台通信』 和 第 7 页的『如何使用跨平台通信示例』。

关于此任务

此示例涵盖在 IBM i 上设置从 IBM MQ 到以下平台的跨平台通信:

- Windows
- AIX
- Linux
- z/OS 或 MVS
- VSE/ESA

过程

1. 使用下列其中一个选项建立网络连接。
   - 建立 LU 6.2 连接，如 第 14 页的『在 IBM i 上建立 LU 6.2 连接』中所述。
   - 建立 TCP 连接，如 第 22 页的『在 IBM i 上建立 TCP 连接』中所述。
2. 建立连接后，定义一些通道，如 第 24 页的『在 IBM i 上配置通道』中所述。

IBM i 在 IBM i 上建立 LU 6.2 连接

要在 IBM i 上建立 LU 6.2 连接，需要配置本地节点并将其连接到伙伴节点。

关于此任务

有关设置从 IBM i 系统到其他某个 IBM MQ 平台的通信所需的参数的更多信息，请参阅 第 18 页的『IBM i 上的 LU 6.2 连接的配置参数』中的表。 任务步骤中括号 () 中的数字对应于这些表的 标识 列中的值。

要配置本地节点，您需要:

- 创建线路描述
- 添加路由项，然后启动子系统

要连接到伙伴节点，您需要:

- 创建控制器描述
- 创建设备描述
- 创建 CPI-C 辅助信息
- 为 APPC 添加通信条目

- 添加配置列表条目

过程

1. 通过创建线路描述并添加路由条目来配置本地节点。
   a) 创建线路描述。
   如果尚未创建线路描述，请使用 **CRTLINTRN** 命令为 **线路描述** (6) 和 **资源名称** (7) 指定值，如以下示例中所示:

```
Create Line Desc (token-ring) (CRTLINTRN)

Type choices, press Enter.

Line description . . . . . .  TOKENRINGL  Name
Resource name . . . . . .  LIN041    Name, *NWID
NWI type . . . . . .  *FR      *FR, *ATM
Online at IPL . . . . . .  *YES     *YES, *NO
Vary on wait . . . . . .  *NOWAIT    *NOWAIT, 15-180 (1 second)
Maximum controllers . . . . . .  40      1-256
Attached NWI . . . . . .  *NONE     Name, *NONE



Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display    F24=More keys
Parameter LIND required.                          +
```

   b) 添加路由项。
   输入命令 **ADDRTGE** 并按 Enter 键，然后为 **子系统描述** (5) 指定您自己的值，并为以下示例中显示的值指定 **路由条目序号**， **比较值** (8)， **起始位置**， **要调用的程序**以及包含要调用的程序的 **库** 。

```
Add Routing Entry (ADDRTGE)

Type choices, press Enter.

Subsystem description . . . . .  QCMN     Name
Library . . . . . .  *LIBL    Name, *LIBL, *CURLIB
Routing entry sequence number .  1       1-9999
Comparison data:
Compare value . . . . . .  'MQSERIES'

Starting position . . . . . .  37      1-80
Program to call . . . . . .  AMQCRC6B   Name, *RTGDTA
Library . . . . . .  QMAS400    Name, * LI BL, *CURLIB
Class . . . . . .  *SBSD     Name, *SBSD
Library . . . . . .  *LIBL     Name, *LIBL, *CURLIB
Maximum active routing steps . .  *NOMAX    0-1000, *NOMAX
Storage pool identifier . . . .  1       1-10



Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
Parameter SBSD required.                          +
```

   通过输入命令 STRSBS *subsystem description* (5) 并按 Enter 键来启动子系统。
2. 通过创建控制器描述，设备描述和 CPI-C 端信息，并为 APPC 添加通信项和配置列表项，创建与伙伴节点的连接。

   Windows 此示例适用于与 Windows 系统的连接，但其他节点的步骤相同。

   a) 创建控制器描述
   在命令行类型 CRTCTLAPPC 上，按 Enter 键，然后指定 **控制器描述** (12) 的值，将 **链接类型** 设置为 *LAN，并将 **IPL 时联机** 设置为 *NO。

```
Create Ctl Desc (APPC) (CRTCTLAPPC)

Type choices, press Enter.

Controller description . . . . .  WINNTCP     Name
Link type . . . . . . . . . . . .  *LAN     *FAX, *FR, *IDLC,
*LAN...
Online at IPL . . . . . . . . . .  *NO      *YES, *NO



Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display    F24=More keys
Parameter CTLD required.                          +
```

按两次 Enter 键，后跟 F10，然后指定 **交换线路列表** (6)， **远程网络标识** (9)， **远程控制点** (10) 和 **LAN 远程适配器地址** (16) 的值并按 Enter 键。

```
Create Ctl Desc (APPC) (CRTCTLAPPC)

Type choices, press Enter.

Controller description . . . . . > WINNTCP     Name
Link type . . . . . . . . . . . . > *LAN     *FAX, *FR, *IDLC, *LAN...
Online at IPL . . . . . . . . . . > *NO      *YES, *NO
APPN-capable . . . . . . . . . .  *YES     *YES, *NO
Switched line list . . . . . . .  TOKENRINGL  Name
+ for more values
Maximum frame size . . . . . . .  *LINKTYPE   265-16393, 256, 265, 512...
Remote network identifier . . .  NETID     Name, *NETATR, *NONE, *ANY
Remote control point . . . . . .  WINNTCP     Name, *ANY
Exchange identifier . . . . . .         00000000-FFFFFFFF
Initial connection . . . . . . .  *DIAL     *DIAL, *ANS
Dial initiation . . . . . . . .  *LINKTYPE   *LINKTYPE, *IMMED, *DELAY
LAN remote adapter address . . .  10005AFC5D83 000000000001-FFFFFFFFFFFF
APPN CP session support . . . .  *YES     *YES, *NO
APPN node type . . . . . . . . .  *ENDNODE   *ENDNODE, *LENNODE...
APPN transmission group number   1       1-20, *CALC
More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

b) 创建设备描述。

输入命令 CRTDEVAPPC 并按 Enter 键，然后指定 **设备描述** (13)， **远程位置** (11)， **本地位置** (3)， **远程网络标识** (9) 和 **连接的控制器** (12) 的值。

```
Create Device Desc (APPC) (CRTDEVAPPC)

Type choices, press Enter.

Device description . . . . . .  WINNTLU     Name
Remote location . . . . . .  WINNTLU     Name
Online at IPL . . . . . .  *YES     *YES, *NO
Local location . . . . . .  AS400LU    Name, *NETATR
Remote network identifier . . .  NETID     Name, *NETATR, *NONE
Attached controller . . . . . .  WINNTCP     Name
Mode . . . . . .  *NETATR    Name, *NETATR
+ for more values
Message queue . . . . . .  QSYSOPR    Name, QSYSOPR
Library . . . . . .  *LIBL    Name, *LIBL, *CURLIB
APPN-capable . . . . . .  *YES     *YES, *NO
Single session:
Single session capable . . . .  *NO      *NO, *YES
Number of conversations . . .         1-512


Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display    F24=More keys
Parameter DEVD required.                          +
```

注: 您可以通过利用 IBM i 自动配置服务来避免手动创建控制器和设备描述。 有关更多信息，请参阅 IBM i 文档。

c) 创建 CPI-C 辅助信息。

输入 CRTCSI 并按 F10，然后指定 **辅助信息** (14)， **远程位置** (11)， **事务程序** (15)， **本地位置** (3)， **方式**和 **远程网络标识** (9) 的值，然后按 Enter 键。

```
Create Comm Side Information (CRTCSI)

Type choices, press Enter.

Side information . . . . . .  NTCPIC    Name
Library . . . . . .  *CURLIB   Name, *CURLIB
Remote location . . . . . .  WINNTLU     Name
Transaction program . . . . . .  MQSERIES

Text 'description' . . . . . .  *BLANK


Additional Parameters

Device . . . . . .  *LOC     Name, *LOC
Local location . . . . . .  AS400LU    Name, *LOC, *NETATR
Mode . . . . . .  #INTER    Name, *NETATR
Remote network identifier . . .  NETID     Name, *LOC, *NETATR, *NONE
Authority . . . . . .  *LIBCRTAUT  Name, *LIBCRTAUT, *CHANGE...

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
Parameter CSI required.
```

d) 为 APPC 添加通信项。

在命令行中，输入 ADDCMNE 并按 Enter 键，然后指定 **子系统描述** (5) 和 **设备** (13) 的值，然后再次按 Enter 键。

```
Add Communications Entry (ADDCMNE)

Type choices, press Enter.

Subsystem description . . . . .  QCMN      Name
Library . . . . . .  *LIBL     Name, *LIBL, *CURLIB
Device . . . . . .  WINNTLU      Name, generic*, *ALL...
Remote location . . . . . .  Name
Job description . . . . . .  *USRPRF     Name, *USRPRF, *SBSD
Library . . . . . .  Name, *LIBL, *CURLIB
Default user profile . . . . . .  *NONE      Name, *NONE, *SYS
Mode . . . . . .  *ANY      Name, *ANY
Maximum active jobs . . . . . .  *NOMAX     0-1000, *NOMAX



Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
Parameter SBSD required.
```

e) 添加配置列表条目。

输入 ADDCFGLE *APPNRMT 并按 F4，然后指定 **远程位置名** (11)， **远程网络标识** (9)， **本地位置名** (3)， **远程控制点** (10) 和 **控制点网络标识** (9) 的值，然后按 Enter 键。

```
Add Configuration List Entries (ADDCFGLE)

Type choices, press Enter.

Configuration list type . . . . > *APPNRMT   *APPNLCL, *APPNRMT...
APPN remote location entry:
Remote location name . . . . .  WINNTLU      Name, generic*, *ANY
Remote network identifier . .  NETID      Name, *NETATR, *NONE
Local location name . . . . .  AS400LU     Name, *NETATR
Remote control point . . . . .  WINNTCP     Name, *NONE
Control point net ID . . . . .  NETID      Name, *NETATR, *NONE
Location password . . . . . .  *NONE
Secure location . . . . . .  *NO       *YES, *NO
Single session . . . . . .  *NO       *YES, *NO
Locally controlled session . .  *NO       *YES, *NO
Pre-established session . . .  *NO       *YES, *NO
Entry 'description' . . . . .  *BLANK
Number of conversations . . .  10       1-512
+ for more values

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

下一步做什么

现在已建立 LU 6.2 连接。 您已准备好完成配置，如 第 24 页的『在 IBM i 上配置通道』中所述。

相关任务

第 22 页的『在 IBM i 上建立 TCP 连接』
如果已配置 TCP，那么没有额外的配置任务。 如果未配置 TCP/IP，那么需要添加 TCP/IP 接口，添加 TCP/IP 回送接口并添加缺省路由。

IBM i *IBM i 上的 LU 6.2 连接的配置参数*

使用 LU 6.2 连接在 IBM i 系统上设置从 IBM MQ 到其他某个 IBM MQ 平台的通信所需的参数。

将这些表与要连接到的平台的表配合使用。

其中数字出现在 参考 列中，它们指示值必须与本节中其他位置的相应表中的值相匹配。 第 14 页的『在 IBM i 上建立 LU 6.2 连接』中的任务步骤引用此表的 标识 列中的值。

在 第 21 页的『表中使用的术语的说明』中说明了 参数名称 列中的条目。

本地节点的定义

表 7: 本地节点定义的配置示例			
标识	**参数名**	**参考**	**使用的示例**
1	本地网络标识		**网络标识**
2	本地控制点名称		**AS400PU**
3	LU 名		**AS400LU**
4	LAN 目标地址		**10005A5962EF**
5	子系统描述		**QCMN**
6	线路描述		**托肯灵**
7	资源名称		**LIN041**
8	本地事务程序名		**MQSERIES**

连接到 Windows 上的 IBM MQ

Windows

Windows

表 8: 用于连接到 *Windows* 上的 *IBM MQ* 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
9	网络标识	2	**网络标识**
10	控制点名称	3	**WINNTCP**
11	LU 名	5	**LU**
12	控制器描述		**WINNTCP**
13	设备		**LU**
14	辅助信息		**NTCPIC**
15	事务程序	7	**MQSERIES**
16	LAN 适配器地址	9	**08005AA5FAB9**
17	方式	17	**#INTER**

连接到 AIX 上的 IBM MQ

AIX

表 9: 用于连接到 *AIX* 系统上的 *IBM MQ* 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
9	网络标识	1	**网络标识**
10	控制点名称	2	**AIXPU**
11	LU 名	4	**AIX 鲁**
12	控制器描述		**AIXPU**
13	设备		**AIX 鲁**
14	辅助信息		**AIXCPIC**

表 9: 用于连接到 AIX 系统上的 IBM MQ 的配置示例 (继续)			
标识	**参数名**	**参考**	**使用的示例**
15	事务程序	6	**MQSERIES**
16	LAN 适配器地址	8	**123456789012**
17	方式	14	**#INTER**

连接到 Linux (x86 平台) 上的 IBM MQ

Linux

表 10: 用于在 Linux (x86 平台) 上连接到 IBM MQ 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
9	网络标识	4	**网络标识**
10	控制点名称	2	**联 XPU**
11	LU 名	5	**LINUXLU**
12	控制器描述		**联 XPU**
13	设备		**LINUXLU**
14	辅助信息		**LXCPIC**
15	事务程序	7	**MQSERIES**
16	LAN 适配器地址	8	**08005AC6DF33**
17	方式	6	**#INTER**

连接到 IBM MQ for z/OS

z/OS

表 11: 连接到 IBM MQ for z/OS 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
9	网络标识	2	**网络标识**
10	控制点名称	3	**MVSPU**
11	LU 名	4	**MVSLU**
12	控制器描述		**MVSPU**
13	设备		**MVSLU**
14	辅助信息		**MVSCPIC**
15	事务程序	7	**MQSERIES**
16	LAN 适配器地址	8	**400074511092**
17	方式	6	**#INTER**

与 VSE/ESA 系统的连接

表 *12:* 用于连接到 *VSE/ESA* 系统的配置示例			
标识	**参数名称**	**参考**	**使用的示例**
9	网络标识	1	**网络标识**
10	控制点名称	2	**VSEPU**
11	LU 名	3	**VSELU**
12	控制器描述		**VSEPU**
13	设备		**VSELU**
14	辅助信息		**VSECPIC**
15	事务程序	4	**MQ01**
16	LAN 适配器地址	5	**400074511092**
17	方式		**#INTER**

表中使用的术语的说明

1 2 3
有关如何查找配置的值的信息，请参阅 第 21 页的『如何查找网络属性』。

4 LAN 目标地址
IBM i 系统令牌环适配器的硬件地址。 您可以使用命令 DSPLIND *Line description* (6) 来查找值。

5 子系统描述
此参数是使用队列管理器时处于活动状态的任何 IBM i 子系统的名称。 已使用名称 QCMN，因为它是 IBM i 通信子系统。

6 线路描述
如果已指定此参数，那么将在资源资源名称的 "描述" 字段中指示此参数。 有关更多信息，请参阅第 22 页的『如何查找资源名称的值』。 如果未指定值，那么需要创建线路描述。

7 资源名称
有关如何查找已配置值的信息，请参阅 第 22 页的『如何查找资源名称的值』。

8 本地事务程序名称
尝试与此工作站进行转换的 IBM MQ 应用程序指定要在接收端运行的程序的符号名称。 此名称是在发送方的通道定义上定义的。 为简单起见，请尽可能使用事务程序名 MQSERIES，或者在连接到 VSE/ESA(长度限制为 4 字节) 的情况下，使用 MQTP。

有关更多信息，请参阅 远程队列管理器平台的本地 IBM i 系统上的设置。

12 控制器描述
此参数是伙伴系统的控制点名称 (或节点名) 的别名。 为了方便起见，我们在此示例中使用了合作伙伴的实际名称。

13 设备
此参数是伙伴系统的 LU 的别名。 为了方便起见，我们在此示例中使用了合作伙伴的 LU 名。

14 辅助信息
此参数是提供给 CPI-C 端信息概要文件的名称。 指定您自己的 8 字符名称。

如何查找网络属性

本地节点已部分配置为 IBM i 安装的一部分。 要显示当前网络属性，请输入命令 **DSPNETA**。

如果需要更改这些值，请使用命令 **CHGNETA**。 可能需要 IPL 才能应用更改。

```
Display Network Attributes
System:  AS400PU
Current system name . . . . . . . . . . . . . . . :  AS400PU
Pending system name . . . . . . . . . . . . . . . :
Local network ID . . . . . . . . . . . . . . . . . :  NETID
Local control point name . . . . . . . . . . . . . :  AS400PU
Default local location . . . . . . . . . . . . . . :  AS400LU
Default mode . . . . . . . . . . . . . . . . . . . :  BLANK
APPN node type . . . . . . . . . . . . . . . . . . :  *ENDNODE
Data compression . . . . . . . . . . . . . . . . . :  *NONE
Intermediate data compression . . . . . . . . . . :  *NONE
Maximum number of intermediate sessions . . . . . :  200
Route addition resistance . . . . . . . . . . . . :  128
Server network ID/control point name . . . . . . . :  NETID    NETCP




More...
Press Enter to continue.

F3=Exit  F12=Cancel
```

检查 **本地网络标识** (1)， **本地控制点名称** (2) 和 **缺省本地位置** (3) 的值是否与表中的值或您自己的值 (如果已更改) 相对应。

如何查找资源名称的值

要查找资源名称的值，请输入 WRKHDWRSC TYPE(*CMN) 并按 Enter 键。

此时将显示 "使用通信资源" 面板。 找到 **资源名称** 的值作为令牌环端口。 在此示例中为 LIN041 。

```
Work with Communication Resources
System:  AS400PU
Type options, press Enter.
2=Edit  4=Remove  5=Work with configuration description
7=Add configuration description ...

Configuration
Opt Resource     Description  Type Description
CC02              2636 Comm Processor
LIN04            2636 LAN Adapter
LIN041    TOKEN-RING  2636 Token-ring Port









Bottom
F3=Exit   F5=Refresh  F6=Print  F11=Display resource addresses/statuses
F12=Cancel  F23=More options
```

IBM i 在 IBM i 上建立 TCP 连接

如果已配置 TCP ，那么没有额外的配置任务。 如果未配置 TCP/IP ，那么需要添加 TCP/IP 接口，添加 TCP/IP 回送接口并添加缺省路由。

过程

1. 添加 TCP/IP 接口。

在命令行中，输入 ADDTCPIFC 并按 Enter 键，然后指定 **IP 地址** 和 **线路描述**以及机器的 **子网掩码** ，然后再次按 Enter 键。

```
Add TCP/IP Interface (ADDTCPIFC)

Type choices, press Enter.

Internet address . . . . . .  19.22.11.55
Line description . . . . . .  TOKENRINGL  Name, *LOOPBACK
Subnet mask . . . . . .  255.255.0.0
Type of service . . . . . .  *NORMAL    *MINDELAY, *MAXTHRPUT..
Maximum transmission unit . . .  *LIND     576-16388, *LIND
Autostart . . . . . .  *YES     *YES, *NO
PVC logical channel identifier          001-FFF
+ for more values
X.25 idle circuit timeout . . .  60      1-600
X.25 maximum virtual circuits .  64      0-64
X.25 DDN interface . . . . . .  *NO      *YES, *NO
TRLAN bit sequencing . . . . . .  *MSB     *MSB, *LSB



Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

2. 添加 TCP/IP 回送接口。

   在命令行中，输入 ADDTCPIFC 并按 Enter 键，然后指定 **IP 地址**， **线路描述**和 **子网掩码**的值。

```
Add TCP Interface (ADDTCPIFC)

Type choices, press Enter.

Internet address . . . . . .  127.0.0.1
Line description . . . . . .  *LOOPBACK   Name, *LOOPBACK
Subnet mask . . . . . .  255.0.0.0
Type of service . . . . . .  *NORMAL    *MINDELAY, *MAXTHRPUT..
Maximum transmission unit . . .  *LIND     576-16388, *LIND
Autostart . . . . . .  *YES     *YES, *NO
PVC logical channel identifier          001-FFF
+ for more values
X.25 idle circuit timeout . . .  60      1-600
X.25 maximum virtual circuits .  64      0-64
X.25 DDN interface . . . . . .  *NO      *YES, *NO
TRLAN bit sequencing . . . . . .  *MSB     *MSB, *LSB



Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

3. 添加缺省路由。

   在命令行中，输入 ADDTCPRTE 并按 Enter 键，然后指定适合于您的网络的值，然后按 Enter 键以创建缺省路由条目。

```
Add TCP Route (ADDTCPRTE)

Type choices, press Enter.

Route destination . . . . . .  *DFTROUTE
Subnet mask . . . . . .  *NONE
Type of service . . . . . .  *NORMAL    *MINDELAY, *MAXTHRPUT.
Next hop . . . . . .  19.2.3.4
Maximum transmission unit . . .  576      576-16388, *IFC



Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
Command prompting ended when user pressed F12.
```

下一步做什么

现在已建立 TCP 连接。 您已准备好完成配置，如 第 24 页的『在 IBM i 上配置通道』中所述。

相关任务

第 14 页的『在 IBM i 上建立 LU 6.2 连接』
要在 IBM i 上建立 LU 6.2 连接，需要配置本地节点并将其连接到伙伴节点。

IBM i 在 IBM i 上配置通道

要为 IBM i 上的示例配置配置 IBM MQ ， 请完成队列管理器的基本配置步骤，然后配置发送方和接收方通道。

关于此任务

使用 **WRKMQMQ** 命令可显示 IBM MQ 配置菜单。

使用命令 **STRMQMLSR** 启动 TCP 通道侦听器。

使用命令 STRMQMCHL CHLNAME(*channel_name*)启动任何发送方通道。

注: AMQ* 错误将放入与发现错误的作业相关的日志中。 使用 **WRKACTJOB** 命令可显示作业列表。 在子系统名称 QSYSWRK 下，找到该作业并针对它输入 5 以使用该作业。 IBM MQ 日志以 AMQ 为前缀。

过程

1. 创建队列管理器。
   a) 输入 CRTMQM 并按 Enter 键。

```
                    Create Message Queue Manager (CRTMQM)

Type choices, press Enter.

Message Queue Manager name . . .

Text 'description' . . . . . .   *BLANK

Trigger interval . . . . . .   999999999     0-999999999
Undelivered message queue  . . .   *NONE

Default transmission queue . . .   *NONE

Maximum handle limit . . . . . .   256           1-999999999
Maximum uncommitted messages . .   1000          1-10000
Default Queue manager  . . . . .   *NO           *YES, *NO




                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

b) 在 **消息队列管理器名称** 字段中，输入 AS400，在 **未传递的消息队列** 字段中，输入 DEAD.LETTER.QUEUE，然后按 Enter 键。

c) 通过输入 STRMQM MQMNAME(AS400)来启动队列管理器。

d) 使用以下参数创建未传递的消息队列:

```
  Local Queue
       Queue name :   DEAD.LETTER.QUEUE
       Queue type :   *LCL
```

有关如何定义队列的更多信息和示例，请参阅步骤 第 25 页的『2』。

2. 定义队列。

a) 在命令行上输入 CRTMQMQ。

```
Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Queue name . . . . . .

Queue type . . . . . .         *ALS, *LCL, *RMT









Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
Parameter QNAME required.
```

b) 填写此面板的两个字段，然后按 Enter 键。

然后将显示另一个面板，其中包含您拥有的其他参数的输入字段。 可以对所有其他队列属性采用缺省值。

3. 定义通道。

a) 在命令行上输入 CRTMQMCHL 以显示 **创建 MQM 通道** 面板。

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Channel name . . . . . .
Channel type . . . . . .         *RCVR, *SDR, *SVR, *RQSTR









Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
Parameter CHLNAME required.
```

b) 填写此面板的两个字段，然后按 Enter 键。

然后，将显示另一个面板，您可以在该面板上指定先前给定的其他参数的值。 可以对所有其他通道属性采用缺省值。

4. 配置示例配置的通道。

有关以下示例中使用的参数的更多信息，请参阅 第 27 页的『IBM i 的通道配置参数』。

Windows 这些示例用于将 IBM i 上的 IBM MQ 与 Windows 上的 IBM MQ 连接。 要在另一个平台上连接到 IBM MQ，请使用 第 27 页的『IBM i 的通道配置参数』 中表中的相应值，而不是 Windows 的值。

a) 定义发送方通道，如以下示例中所示:

- 使用 SNA:

```
     Local Queue
              Queue name :   WINNT                           F
              Queue type :   *LCL
                   Usage :   *TMQ

     Remote Queue
              Queue name :   WINNT.REMOTEQ                   D
              Queue type :   *RMT
            Remote queue :   WINNT.LOCALQ                    E
    Remote Queue Manager :   WINNT                           C
      Transmission queue :   WINNT                           F

     Sender Channel
            Channel Name :   AS400.WINNT.SNA                 G
            Channel Type :   *SDR
          Transport type :   *LU62
        Connection name :    WINNTCPIC                       14
       Transmission queue :  WINNT                           F
```

- 使用 TCP:

```
     Local Queue
              Queue name :   WINNT                           F
              Queue type :   *LCL
                   Usage :   *TMQ

     Remote Queue
              Queue name :   WINNT.REMOTEQ                   D
              Queue type :   *RMT
            Remote queue :   WINNT.LOCALQ                    E
    Remote Queue Manager :   WINNT                           C
```

```
 Transmission queue :   WINNT                              F

Sender Channel
       Channel Name :   AS400.WINNT.TCP                    H
       Channel Type :   *SDR
     Transport type :   *TCP
    Connection name :   WINNT.tcpip.hostname
 Transmission queue :   WINNT                              F
```

b) 定义接收方通道，如以下示例中所示:

- 使用 SNA:

```
Local Queue
         Queue name :   AS400.LOCALQ                     B
         Queue type :   *LCL

Receiver Channel
       Channel Name :   WINNT.AS400.SNA                  I
       Channel Type :   *RCVR
     Transport type :   *LU62
```

- 使用 TCP:

```
Local Queue
         Queue name :   AS400.LOCALQ                     B
         Queue type :   *LCL

Receiver Channel
       Channel Name :   WINNT.AS400.TCP                  J
       Channel Type :   *RCVR
     Transport type :   *TCP
```

IBM i *IBM i 的通道配置参数*

为 IBM i 上的示例配置配置通道所需的参数。

步骤 第 26 页的『4』 (共 第 24 页的『在 IBM i 上配置通道』 步) 描述了要在 IBM i 队列管理器上执行的配置，以实现 第 5 页的『示例: 为 IBM MQ 设置跨平台通信』中描述的通道。 第 24 页的『在 IBM i 上配置通道』 中的示例用于连接 IBM MQ for IBM i 和 IBM MQ for Windows。 要在另一个平台上连接到 IBM MQ，请使用相应表中的值来代替 Windows 的值。

注:

1. **粗体** 中的词是建议值，并反映在这些示例中使用的 IBM MQ 对象的名称。 您可以在产品安装中更改这些值，但如果可以，请确保在处理本节中的示例时使用您自己的值。
2. IBM MQ 通道 ping 命令 (**PNGMQMCHL**) 以交互方式运行，而启动通道会导致提交批处理作业。 如果通道 ping 成功完成但通道未启动，那么网络和 IBM MQ 定义可能正确，但批处理作业的 IBM i 环境不正确。例如，确保 QSYS2 包含在库列表的系统部分中，而不仅仅是您的个人库列表中。

有关如何创建表中列出的对象的更多信息和示例，请参阅 第 24 页的『在 IBM i 上配置通道』。

本地节点的定义

表 *13:* 本地节点定义的配置示例

标识	参数名	参考	使用的示例
A	队列管理器名称		**AS400**
B	本地队列名称		**AS400.LOCALQ**

连接到 Windows 上的 IBM MQ

Windows

表的此部分中的值必须与 第 39 页的『Windows 的通道配置参数』中使用的值匹配，如所指示。

表 *14:* 用于连接到 *Windows* 上的 *IBM MQ* 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**winnt**
D	远程队列名称		**WINNT.REMOTEQ**
E	远程系统上的队列名称	B	**WINNT.LOCALQ**
F	传输队列的名称		**winnt**
G	发送方 (SNA) 通道名称		**AS400.WINNT.SNA**
H	发送方 (TCP/IP) 通道名称		**AS400.WINNT.TCP**
I	接收方 (SNA) 通道名称	G	**WINNT.AS400.SNA**
J	接收方 (TCP/IP) 通道名称	H	**WINNT.AS400.TCP**

连接到 AIX 上的 IBM MQ

AIX

表的此部分中的值必须与 第 11 页的『AIX 的通道配置参数』中使用的值匹配，如所指示。

表 *15:* 用于连接到 *AIX* 上的 *IBM MQ* 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**AIX**
D	远程队列名称		**AIX。远程队列**
E	远程系统上的队列名称	B	**AIX.LOCALQ**
F	传输队列的名称		**AIX**
G	发送方 (SNA) 通道名称		**AS400.AIX.SNA**
H	发送方 (TCP/IP) 通道名称		**AS400.AIX。TCP**
I	接收方 (SNA) 通道名称	G	**AIX.AS400.SNA**
J	接收方 (TCP) 通道名称	H	**AIX.AS400.TCP**

连接到 Linux 上的 IBM MQ

Linux

表的此部分中的值必须与 第 33 页的『Linux 的通道配置参数』中使用的值匹配，如所指示。

表 *16:* 用于连接到 *Linux* 上的 *IBM MQ* 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**Linux**
D	远程队列名称		**LINUX.REMOTEQ**
E	远程系统上的队列名称	B	**LINUX.LOCALQ**
F	传输队列的名称		**Linux**
G	发送方 (SNA) 通道名称		**AS400.LINUX.SNA**
H	发送方 (TCP/IP) 通道名称		**AS400.LINUX.TCP**
I	接收方 (SNA) 通道名称	G	**LINUX.AS400.SNA**

表 16: 用于连接到 Linux 上的 IBM MQ 的配置示例 (继续)			
标识	**参数名**	**参考**	**使用的示例**
J	接收方 (TCP/IP) 通道名称	H	**LINUX.AS400.TCP**

连接到 IBM MQ for z/OS

z/OS

表的此部分中的值必须与 第 43 页的『z/OS 的通道配置参数』中使用的值匹配，如所指示。

表 17: 用于连接到 IBM MQ for z/OS 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**MVS**
D	远程队列名称		**MVS.REMOTEQ**
E	远程系统上的队列名称	B	**MVS.LOCALQ**
F	传输队列的名称		**MVS**
G	发送方 (SNA) 通道名称		**AS400.MVS.SNA**
H	发送方 (TCP) 通道名称		**AS400.MVS.TCP**
I	接收方 (SNA) 通道名称	G	**MVS.AS400.SNA**
J	接收方 (TCP) 通道名称	H	**MVS.AS400.TCP**

与 VSE/ESA 系统的连接

表的此部分中的值必须与 VSE/ESA 系统中使用的值匹配。

表 18: 用于连接到 VSE/ESA 系统的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**VSE**
D	远程队列名称		**VSE.REMOTEQ**
E	远程系统上的队列名称	B	**VSE.LOCALQ**
F	传输队列的名称		**VSE**
G	发送方通道名称		**AS400.VSE.SNA**
I	接收方通道名称	G	**VSE.AS400.SNA**

Linux 示例: 在 Linux 上设置 IBM MQ 跨平台通信

此示例显示如何在另一个平台上设置从 IBM MQ on Linux 到 IBM MQ 的通信链路，并建立到该平台的工作通道。

开始之前

有关此示例以及如何使用此示例的背景信息，请参阅 第 5 页的『示例: 为 IBM MQ 设置跨平台通信』 和 第 7 页的『如何使用跨平台通信示例』。

关于此任务

此示例涵盖在 Linux 上设置从 IBM MQ 到以下平台的跨平台通信:

- Windows
- AIX
- IBM i
- z/OS

MQ_INSTALLATION_PATH 表示 IBM MQ 安装所在的高级目录。

使用 LU 6.2 或 TCP 建立网络连接。

注: 对于 TCP，某些 Linux 分发现在使用扩展的 inet 守护程序 (XINETD) 而不是 inet 守护程序 (INETD)。 以下指示信息告诉您如何使用 inet 守护程序或扩展 inet 守护程序建立 TCP 连接。

过程

1. 使用 LU6.2 建立网络连接

   注: 本部分中的信息仅适用于 IBM MQ for Linux (x86 平台)。 它不适用于 IBM MQ for Linux (x86-64 platform)、 IBM MQ for Linux s390x platform) 或 IBM MQ for Linux (Power platform)。

   For the latest information about configuring SNA over TCP/IP, refer to the the Administration Guide for your version of Linux from the following documentation: Linux 库上的 Communications Server for Data Center Deployment.
2. 使用 inet 守护程序 (INETD) 建立 TCP 连接

   a) 编辑文件 `/etc/services`。

   如果文件中没有以下行，请按如下所示进行添加:

   ```
   MQSeries    1414/tcp   # MQSeries channel listener
   ```

   注: 要编辑此文件，您必须以超级用户或 root 用户身份登录。

   b) 编辑文件 `/etc/inetd.conf`。

   如果该文件中没有以下行，请按如下所示添加该文件:

   ```
   MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta
   [-m queue.manager.name ]
   ```

   c) 使用以下命令查找 inetd 的进程标识:

   ```
   ps -ef | grep inetd
   ```

   d) 运行以下命令:

   ```
   kill -1 inetd processid
   ```

   如果系统上有多个队列管理器，因此需要多个服务，那么必须将每个额外队列管理器的行添加到 `/etc/services` 和 `inetd.conf`。

   例如:

   ```
   MQSeries1   1414/tcp
   MQSeries2   1822/tcp
   ```

   ```
   MQSeries1 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM1
   MQSeries2 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM2
   ```

   如果在单个 TCP 端口上排队的未完成连接请求数存在限制，那么这将避免生成错误消息。 有关未完成的连接请求数的信息，请参阅 使用 TCP 侦听器储备选项。

   Linux 上的 inetd 进程可以限制 TCP 端口上的入站连接速率。 缺省值为 60 秒时间间隔内的 40 个连接。 如果需要更高的速率，请在 60 秒时间间隔内指定入站连接数的新限制，方法是在 inetd.conf 中

追加一个句点 (.)，后跟相应服务的 nowait 参数的新限制。 例如，对于 60 秒时间间隔内使用的 500 个连接的限制:

```
MQSeries stream tcp nowait.500 mqm / MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM1
```

MQ_INSTALLATION_PATH 表示 IBM MQ 安装所在的高级目录。

3. 使用扩展的 inet 守护程序 (XINETD) 建立 TCP 连接

以下指示信息描述了如何在 Red Hat Linux 上实现扩展的 inet 守护程序。 如果您正在使用不同的 Linux 分发版，那么可能必须调整这些指示信息。

a) 编辑文件 /etc/services。

如果文件中没有以下行，请按如下所示进行添加:

```
MQSeries    1414/tcp   # MQSeries channel listener
```

如果文件中没有以下行，请按如下所示进行添加:

```
MQSeries    1414/tcp   # MQSeries channel listener
```

b) 通过向文件添加以下节，在 XINETD 配置目录 /etc/xinetd.d 中创建名为 IBM MQ 的文件:

```
# IBM MQ service for XINETD
service MQSeries
{
  disable         = no
  flags           = REUSE
  socket_type     = stream
  wait            = no
  user            = mqm
  server          = MQ_INSTALLATION_PATH/bin/amqcrsta
  server_args     = -m queue.manager.name
  log_on_failure += USERID
}
```

c) 通过发出以下命令重新启动扩展的 inet 守护程序:

```
/etc/rc.d/init.d/xinetd restart
```

如果系统上有多个队列管理器，因此需要多个服务，那么必须为每个额外的队列管理器向 /etc/services 添加一行。 您可以在 /etc/xinetd.d 目录中为每个服务创建一个文件，也可以将其他节添加到先前创建的 IBM MQ 文件。

Linux 上的 xinetd 进程可以限制 TCP 端口上的入站连接速率。 缺省值为 10 秒时间间隔内的 50 个连接。 如果需要更高的速率，请通过在 xinetd 配置文件中指定 "cps" 属性来指定入站连接速率的新限制。 例如，对于 60 秒时间间隔内使用的 500 个连接的限制:

```
cps = 500 60
```

4. 立即完成配置，建立 TCP/IP 连接。

转至 第 31 页的『在 Linux 上配置通道』。

Linux 在 Linux 上配置通道

要为 Linux 上的示例配置配置 IBM MQ，请完成队列管理器的基本配置步骤，然后配置发送方和接收方通道。

开始之前

在开始此过程之前，请确保先创建 mqm 用户标识和 mqm 组，然后设置密码。

使用以下命令启动任何通道:

```
runmqchl -c channel.name
```

关于此任务

注意:

1. 样本程序安装在 *MQ_INSTALLATION_PATH*/samp 中，其中 *MQ_INSTALLATION_PATH* 表示安装 IBM MQ 的高级目录。
2. 错误日志存储在 /var/mqm/qmgrs/ *qmgrname* /个错误中。
3. 使用命令解释器 **runmqsc** 输入管理命令时，行尾的 + 表示下一行是连续行。 确保最后一个参数与连续字符之间有空格。

过程

1. 设置基本配置:

   a) 使用以下命令从 UNIX 提示符创建队列管理器和一组缺省对象:

   ```
   crtmqm -u dlqname -q linux
   ```

   其中:

   linux
   是队列管理器的名称

   -q
   指示这将成为缺省队列管理器

   -u *dlqname*
   指定死信队列的名称

   b) 使用以下命令从 UNIX 提示符启动队列管理器:

   ```
   strmqm linux
   ```

   其中， *linux* 是创建队列管理器时提供给该队列管理器的名称。

2. 配置示例配置的通道。

   有关以下示例中使用的参数的更多信息，请参阅 第 33 页的『Linux 的通道配置参数』。 在每种情况下，示例都显示 MQSC 命令。 从 Linux 命令行启动 **runmqsc** 并依次输入每个命令，或者将这些命令构建到命令文件中。

   Windows 这些示例用于将 Linux 上的 IBM MQ 与 Windows 上的 IBM MQ 连接。 要在另一个平台上连接到 IBM MQ，请使用 第 33 页的『Linux 的通道配置参数』 中表中的相应值，而不是 Windows 的值。

   a) 定义发送方通道，如以下示例中所示:

   - 使用 SNA

   ```
   def ql (WINNT) +                                F
       usage(xmitq) +
       replace

   def qr (WINNT.REMOTEQ) +                        D
       rname(WINNT.LOCALQ) +                       E
       rqmname(WINNT) +                            C
       xmitq(WINNT) +                              F
       replace

   def chl (LINUX.WINNT.SNA) chltype(sdr) +        G
       trptype(lu62) +
       conname('WINNTCPIC') +                      14
       xmitq(WINNT) +                              F
       replace
   ```

   - 使用 TCP

   ```
   def ql (WINNT) +                                F
       usage(xmitq) +
       replace
   ```

```
def qr (WINNT.REMOTEQ) +                          D
    rname(WINNT.LOCALQ) +                         E
    rqmname(WINNT) +                              C
    xmitq(WINNT) +                                F
    replace

def chl (LINUX.WINNT.TCP) chltype(sdr) +          H
    trptype(tcp) +
    conname(remote_tcpip_hostname) +
    xmitq(WINNT) +                                F
    replace
```

b) 定义接收方通道，如以下示例中所示:

- 使用 SNA:

```
def ql (LINUX.LOCALQ) replace                    B

def chl (WINNT.LINUX.SNA) chltype(rcvr) +         I
    trptype(lu62) +
    replace
```

- 使用 TCP:

```
def ql (LINUX.LOCALQ) replace                    B

def chl (WINNT.LINUX.TCP) chltype(rcvr) +         J
    trptype(tcp) +
    replace
```

Linux *Linux* 的通道配置参数

为 Linux 上的示例配置配置通道所需的参数。

步骤 第 32 页的『2』 (共 第 31 页的『在 Linux 上配置通道』 步) 描述了要在 Linux 队列管理器上执行的配置，以实现 第 5 页的『示例: 为 IBM MQ 设置跨平台通信』中描述的通道。 第 31 页的『在 Linux 上配置通道』 中的示例用于连接 IBM MQ for IBM i 和 IBM MQ for Windows。 要在另一个平台上连接到 IBM MQ，请使用相应表中的值来代替 Windows 的值。

注: 粗体 中的词是建议值，并反映在这些示例中使用的 IBM MQ 对象的名称。 您可以在产品安装中更改这些值，但如果更改了这些值，请确保在使用本部分中的示例时使用自己的值

本地节点的定义

表 *19:* 本地节点定义的配置示例

标识	参数名	参考	使用的示例
A	队列管理器名称		**Linux**
B	本地队列名称		**LINUX.LOCALQ**

连接到 Windows 上的 IBM MQ

Windows

表的此部分中的值必须与 第 39 页的『Windows 的通道配置参数』中使用的值匹配，如所指示。

表 *20:* 用于连接到 *Windows* 上的 *IBM MQ* 的配置示例

标识	参数名	参考	使用的示例
C	远程队列管理器名称	A	**winnt**
D	远程队列名称		**WINNT.REMOTEQ**
E	远程系统上的队列名称	B	**WINNT.LOCALQ**
F	传输队列的名称		**winnt**

表 *20:* 用于连接到 *Windows* 上的 *IBM MQ* 的配置示例 *(继续)*			
标识	**参数名**	**参考**	**使用的示例**
G	发送方 (SNA) 通道名称		**LINUX.WINNT.SNA**
H	发送方 (TCP/IP) 通道名称		**LINUX.WINNT.TCP**
I	接收方 (SNA) 通道名称	G	**WINNT.LINUX.SNA**
J	接收方 (TCP) 通道名称	H	**WINNT.LINUX.TCP**

连接到 AIX 上的 IBM MQ

AIX

表的此部分中的值必须与 第 9 页的『在 AIX 上配置通道』中使用的值匹配，如所指示。

表 *21:* 用于连接到 *AIX* 上的 *IBM MQ* 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**AIX**
D	远程队列名称		**AIX.REMOTEQ**
E	远程系统上的队列名称	B	**AIX.LOCALQ**
F	传输队列的名称		**AIX**
G	发送方 (SNA) 通道名称		**.LINUX.AIX.SNA**
H	发送方 (TCP/IP) 通道名称		**LINUX.AIX.TCP**
I	接收方 (SNA) 通道名称	G	**AIX.LINUX.SNA**
J	接收方 (TCP/IP) 通道名称	H	**AIX.LINUX.TCP**

连接到 IBM MQ for IBM i

IBM i

表的此部分中的值必须与 第 27 页的『IBM i 的通道配置参数』中使用的值匹配，如所指示。

表 *22:* 用于连接到 *IBM i* 上的 *IBM MQ* 的配置示例			
标识	**参数名称**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**AS400**
D	远程队列名称		**AS400.REMOTEQ**
E	远程系统上的队列名称	B	**AS400.LOCALQ**
F	传输队列的名称		**AS400**
G	发送方 (SNA) 通道名称		**LINUX.AS400.SNA**
H	发送方 (TCP) 通道名称		**LINUX.AS400.TCP**
I	接收方 (SNA) 通道名称	G	**AS400.LINUX.SNA**
J	接收方 (TCP) 通道名称	H	**AS400.LINUX.TCP**

连接到 IBM MQ for z/OS

z/OS

表的此部分中的值必须与 第 43 页的『z/OS 的通道配置参数』中使用的值匹配，如所指示。

表 *23:* 用于连接到 *IBM MQ for z/OS* 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**MVS**
D	远程队列名称		**MVS.REMOTEQ**
E	远程系统上的队列名称	B	**MVS.LOCALQ**
F	传输队列的名称		**MVS**
G	发送方 (SNA) 通道名称		**LINUX.MVS.SNA**
H	发送方 (TCP) 通道名称		**LINUX.MVS.TCP**
I	接收方 (SNA) 通道名称	G	**MVS.LINUX.SNA**
J	接收方 (TCP) 通道名称	H	**MVS.LINUX.TCP**

使用队列共享组连接到 IBM MQ for z/OS

z/OS

表的此部分中的值必须与 第 51 页的『Shared channel configuration parameters』中使用的值匹配，如所指示。

表 *24:* 使用队列共享组连接到 *IBM MQ for z/OS* 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**QSG**
D	远程队列名称		**QSG.REMOTEQ**
E	远程系统上的队列名称	B	**QSG.SHAREDQ**
F	传输队列的名称		**QSG**
G	发送方 (SNA) 通道名称		**LINUX.QSG.SNA**
H	发送方 (TCP) 通道名称		**LINUX.QSG.TCP**
I	接收方 (SNA) 通道名称	G	**QSG.LINUX.SNA**
J	接收方 (TCP) 通道名称	H	**QSG.LINUX.TCP**

Windows 示例: 在 Windows 上设置 IBM MQ 跨平台通信

此示例显示如何在另一个平台上设置从 IBM MQ on Windows 到 IBM MQ 的通信链路，并建立到该平台的工作通道。

开始之前

有关此示例以及如何使用此示例的背景信息，请参阅 第 5 页的『示例: 为 IBM MQ 设置跨平台通信』 和 第 7 页的『如何使用跨平台通信示例』。

关于此任务

此示例涵盖在 Windows 上设置从 IBM MQ 到以下平台的跨平台通信:

- AIX AIX
- IBM i IBM i

- Linux
- z/OS

过程

1. 使用 LU6.2 建立网络连接。

   有关配置 AnyNet SNA over TCP/IP 的信息，请参阅 AnyNet SNA over TCP/IP 和 Communications Server for Windows 。

2. 使用 TCP 建立网络连接。

   Windows 系统随附的 TCP 堆栈不包含 *inet* 守护程序或等效守护程序。

   必须先显式启动侦听器，然后才能启动任何通道。 它允许接收通道自动启动，以响应来自入站发送通道的请求。

   使用以下命令来启动 IBM MQ TCP 侦听器:

   ```
   runmqlsr -t tcp
   ```

3. 使用 NetBIOS 建立网络连接。

   a) 在通道的每一端，指定要由队列管理器配置文件 qm.ini 中的 IBM MQ 通道进程使用的本地 NetBIOS 名称。

   例如，发送端的 Windows 中的 NETBIOS 节可能类似于以下内容:

   ```
   NETBIOS:
    LocalName=WNTNETB1
   ```

   在接收端，如下所示:

   ```
   NETBIOS:
    LocalName=WNTNETB2
   ```

   每个 IBM MQ 进程都必须使用不同的本地 NetBIOS 名称。 请勿将系统名称用作 NetBIOS 名称，因为 Windows 已使用该名称。

   b) 在通道的每一端，验证系统上正在使用的 LAN 适配器号。

   逻辑适配器号 0 的 IBM MQ for Windows 缺省值是在 Internet Protocol 网络上运行的 NetBIOS 。 要使用本机 NetBIOS ，必须选择逻辑适配器号 1。 请参阅 建立 LAN 适配器号。

   在 Windows 注册表的 NETBIOS 节中指定正确的 LAN 适配器号。 例如:

   ```
   NETBIOS:
    AdapterNum=1
   ```

   c) 因此，发送方通道启动工作，请通过 MQNAME 环境变量指定本地 NetBIOS 名称:

   ```
    SET MQNAME=WNTNETB1I
   ```

   注: 此名称必须唯一。

   d) 在发送端，定义一个指定在通道另一端使用的 NetBIOS 名称的通道。

   例如:

   ```
   DEFINE CHANNEL (WINNT.OS2.NET) CHLTYPE(SDR) +
          TRPTYPE(NETBIOS) +
          CONNAME(WNTNETB2) +
          XMITQ(OS2) +
          MCATYPE(THREAD) +
          REPLACE
   ```

   必须指定选项 MCATYPE(THREAD)，因为在 Windows 上，发送方通道必须作为线程运行。

   e) 在接收端，定义相应的接收机信道。

   例如:

```
DEFINE CHANNEL (WINNT.OS2.NET) CHLTYPE(RCVR) +
       TRPTYPE(NETBIOS) +
       REPLACE
```

f) 启动通道启动程序。

每个新通道都作为一个线程启动，而不是作为一个新进程启动:

```
runmqchi
```

g) 在接收端，启动 IBM MQ 侦听器:

```
runmqlsr -t netbios
```

(可选) 可以指定队列管理器名称， NetBIOS 局部名，会话数，名称数和命令数的值。 请参阅 在 Windows 上定义 NetBIOS 连接，以获取有关设置 NetBIOS 连接的更多信息。

4. 立即建立网络连接，完成配置。 请参阅 第 37 页的『在 Windows 上配置通道』。

Windows 在 Windows 上配置通道

要为 Windows 上的示例配置配置 IBM MQ ， 请完成队列管理器的基本配置步骤，然后配置发送方和接收方通道。

关于此任务

注意:

1. 您可以使用样本程序 AMQSBCG 来显示队列中所有消息的内容和头。 例如:

```
AMQSBCG q_name qmgr_name
```

显示队列管理器 *qmgr_name* 中定义的队列 *q_name* 的内容。

或者，可以在 IBM MQ Explorer 中使用消息浏览器。

2. 您可以使用命令从命令提示符启动任何通道

```
runmqchl -c channel.name
```

3. 可以在目录 *MQ_INSTALLATION_PATH*\qmgrs\ *qmgrname* \errors 和 *MQ_INSTALLATION_PATH*\qmgrs\@system\errors 中找到错误日志。 在这两种情况下，最新的消息都位于 amqerr01.log 的末尾。

MQ_INSTALLATION_PATH 表示 IBM MQ 安装所在的高级目录。

4. 使用命令解释器 **runmqsc** 输入管理命令时，行尾的 + 表示下一行是连续行。 确保最后一个参数与连续字符之间有空格。

过程

1. 要使用命令提示符设置基本配置，请完成以下步骤:

a) 使用以下命令创建队列管理器和一组缺省对象:

```
crtmqm -u dlqname -q winnt
```

其中:

winnt
是队列管理器的名称

-q
指示这将成为缺省队列管理器

-u *dlqname*
指定无法传递的消息队列的名称

b) 使用以下命令启动队列管理器:

```
strmqm winnt
```

其中， *winnt* 是创建队列管理器时提供给队列管理器的名称。

2. 配置示例配置的通道。

有关以下示例中使用的参数的更多信息，请参阅 第 39 页的『Windows 的通道配置参数』。 在每种情况下，示例都显示 MQSC 命令。 从 Linux 命令行启动 **runmqsc** 并依次输入每个命令，或者将这些命令构建到命令文件中。 这些示例用于连接 IBM MQ for Windows 和 IBM MQ for AIX。 要在另一个平台上连接到 IBM MQ ，请使用 第 39 页的『Windows 的通道配置参数』 中表中的相应值，而不是 IBM MQ for AIX 的值。

a) 定义发送方通道，如以下示例中所示:

- 使用 SNA

```
def ql (AIX) +                                  F
    usage(xmitq) +
    replace

def qr (AIX.REMOTEQ) +                          D
    rname(AIX.LOCALQ) +                         E
    rqmname(AIX) +                              C
    xmitq(AIX) +                                F
    replace

def chl (WINNT.AIX.SNA) chltype(sdr) +          G
    trptype(lu62) +
    conname(AIXCPIC) +                          18
    xmitq(AIX) +                                F
    replace
```

- 使用 TCP

```
def ql (AIX) +                                  F
    usage(xmitq) +
    replace

def qr (AIX.REMOTEQ) +                          D
    rname(AIX.LOCALQ) +                         E
    rqmname(AIX) +                              C
    xmitq(AIX) +                                F
    replace

def chl (WINNT.AIX.TCP) chltype(sdr) +          H
    trptype(tcp) +
    conname(remote_tcpip_hostname) +
    xmitq(AIX) +                                F
    replace
```

b) 定义接收方通道，如以下示例中所示:

- 使用 SNA:

```
def ql (WINNT.LOCALQ) replace                   B

def chl (AIX.WINNT.SNA) chltype(rcvr) +         I
    trptype(lu62) +
    replace
```

- 使用 TCP:

```
def ql (WINNT.LOCALQ) replace                   B

def chl (AIX.WINNT.TCP) chltype(rcvr) +         J
    trptype(tcp) +
    replace
```

下一步做什么

自动启动

IBM MQ for Windows 允许您自动启动队列管理器及其通道启动程序，通道，侦听器和命令服务器。

使用 IBM MQ 服务插件来定义队列管理器的服务。 成功完成通信设置测试后，请在插件中将相关服务设置为 **automatic** 。 当系统启动时，提供的 IBM MQ 服务可以读取此文件。

有关更多信息，请参阅管理 IBM MQ。

将通道作为进程或线程运行

IBM MQ for Windows 提供了作为 Windows 进程或 Windows 线程运行发送通道的灵活性。 这是在发送方通道定义上的 MCATYPE 参数中指定的。

大多数安装将其发送通道作为线程运行，因为支持许多并发通道连接所需的虚拟内存和实内存会减少。 但是，对于发送消息通道代理程序， NetBIOS 连接需要单独的进程。

Windows *Windows* 的通道配置参数

为 Windows 上的示例配置配置通道所需的参数。

步骤 第 38 页的『2』 (共 第 37 页的『在 Windows 上配置通道』 步) 描述了要在 Linux 队列管理器上执行的配置，以实现 第 5 页的『示例: 为 IBM MQ 设置跨平台通信』中描述的通道。 第 37 页的『在 Windows 上配置通道』 中的示例用于连接 IBM MQ for Windows 和 IBM MQ for AIX。 要在另一个平台上连接到 IBM MQ ，请使用相应表中的值来代替 Windows 的值。

注: 粗体 中的词是建议值，并反映在这些示例中使用的 IBM MQ 对象的名称。 您可以在产品安装中更改这些值，但如果可以，请确保在处理本节中的示例时使用您自己的值。

在每种情况下都会显示 MQSC 命令。 从命令提示符启动 **runmqsc** 并依次输入每个命令，或者将命令构建到命令文件中。

提供了用于连接 IBM MQ for Windows 和 IBM MQ for AIX 的示例。 要在另一个平台上连接到 IBM MQ ，请使用表中的相应值集来代替 Windows 的值集。

本地节点的定义

表 *25:* 本地节点定义的配置示例

标识	参数名	参考	使用的示例
A	队列管理器名称		**winnt**
B	本地队列名称		**WINNT.LOCALQ**

连接到 AIX 上的 IBM MQ

AIX

表的此部分中的值必须与 第 11 页的『AIX 的通道配置参数』中使用的值匹配，如所指示。

表 *26:* 用于连接到 *AIX* 上的 *IBM MQ* 的配置示例

	参数名称	参考	使用的示例
C	远程队列管理器名称	A	**AIX**
D	远程队列名称		**AIX**。远程队列
E	远程系统上的队列名称	B	**AIX.LOCALQ**
F	传输队列的名称		**AIX**
G	发送方 (SNA) 通道名称		**WINNT.AIX.SNA**
H	发送方 (TCP) 通道名称		**WINNT.AIX.TCP**

表 26: 用于连接到 AIX 上的 IBM MQ 的配置示例 (继续)			
	参数名称	**参考**	**使用的示例**
I	接收方 (SNA) 通道名称	G	**AIX.WINNT.SNA**
J	接收方 (TCP) 通道名称	H	**AIX.WINNT.TCP**

连接到 IBM i 上的 IBM MQ

IBM i

表的此部分中的值必须与 第 27 页的『IBM i 的通道配置参数』中使用的值匹配，如所指示。

表 27: 用于连接到 IBM i 上的 IBM MQ 的配置示例			
标识	**参数名称**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**AS400**
D	远程队列名称		**AS400.REMOTEQ**
E	远程系统上的队列名称	B	**AS400.LOCALQ**
F	传输队列的名称		**AS400**
G	发送方 (SNA) 通道名称		**WINNT.AS400.SNA**
H	发送方 (TCP) 通道名称		**WINNT.AS400.TCP**
I	接收方 (SNA) 通道名称	G	**AS400.WINNT.SNA**
J	接收方 (TCP) 通道名称	H	**AS400.WINNT.TCP**

连接到 IBM MQ for z/OS

z/OS

表的此部分中的值必须与 第 43 页的『z/OS 的通道配置参数』中使用的值匹配，如所指示。

表 28: 用于连接到 IBM MQ for z/OS 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**MVS**
D	远程队列名称		**MVS.REMOTEQ**
E	远程系统上的队列名称	B	**MVS.LOCALQ**
F	传输队列的名称		**MVS**
G	发送方 (SNA) 通道名称		**WINNT.MVS.SNA**
H	发送方 (TCP) 通道名称		**WINNT.MVS.TCP**
I	接收方 (SNA) 通道名称	G	**MVS.WINNT.SNA**
J	接收方 (TCP) 通道名称	H	**MVS.WINNT.TCP**

使用队列共享组连接到 IBM MQ for z/OS

z/OS

表的此部分中的值必须与 第 51 页的『Shared channel configuration parameters』中使用的值匹配，如所指示。

表 29: 使用队列共享组连接到 *IBM MQ for z/OS* 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**QSG**
D	远程队列名称		**QSG.REMOTEQ**
E	远程系统上的队列名称	B	**QSG.SHAREDQ**
F	传输队列的名称		**QSG**
G	发送方 (SNA) 通道名称		**WINNT.QSG.SNA**
H	发送方 (TCP) 通道名称		**WINNT.QSG.TCP**
I	接收方 (SNA) 通道名称	G	**QSG.WINNT.SNA**
J	接收方 (TCP) 通道名称	H	**QSG.WINNT.TCP**

z/OS 示例: 在 z/OS 上设置 IBM MQ 跨平台通信

此示例显示如何在另一个平台上设置从 IBM MQ on z/OS 到 IBM MQ 的通信链路，并建立到该平台的工作通道。

开始之前

有关此示例以及如何使用此示例的背景信息，请参阅 第 5 页的『示例: 为 IBM MQ 设置跨平台通信』 和 第 7 页的『如何使用跨平台通信示例』。

关于此任务

此示例涵盖在 z/OS 上设置从 IBM MQ 到以下平台的跨平台通信:

- Windows Windows
- AIX AIX
- Linux Linux
- IBM i IBM i
- VSE/ESA

您还可以连接以下任一项:

- z/OS 到 z/OS
- z/OS 到 MVS
- MVS 到 MVS

过程

1. 建立网络连接。
   - 建立 LU 6.2 连接

     有关配置基于 TCP/IP 的 SNA 的最新信息，请参阅以下联机 IBM 文档: Communications Server for z/OS。
   - 建立 TCP 连接

     使用以下命令更改队列管理器对象以使用正确的分布式排队参数。 必须将 TCP 地址空间的名称添加到 TCPNAME 队列管理器属性。

     ```
     ALTER QMGR TCPNAME(TCPIP)
     ```

现在已建立 TCP 连接。 您已准备好完成配置。

2. 配置通道。

   有关如何配置通道的详细信息，请参阅 第 42 页的『在 IBM MQ for z/OS 上配置通道』 。

z/OS 在 IBM MQ for z/OS 上配置通道

要为 z/OS 上的示例配置配置 IBM MQ ， 请启动并配置通道和侦听器。

过程

1. 使用以下命令启动通道启动程序:

```
/cpf   START CHINIT  1
```

2. 使用以下命令启动 LU 6.2 侦听器:

```
/cpf   START LSTR LUNAME( M1 ) TRPTYPE(LU62)
```

   M1 的 LUNAME 是指您为 LU 提供的符号名称 (5)。 必须指定 TRPTYPE (LU62) ， 否则侦听器将假定您需要 TCP。

3. 使用以下命令启动 TCP 侦听器:

```
/cpf   START LSTR
```

   如果要使用 1414 以外的端口 (缺省 IBM MQ 端口) ， 请使用以下命令:

```
/cpf   START LSTR PORT( 1555 )
```

   如果通道协商检测到消息序号在每一端都不同，那么 IBM MQ 通道不会成功初始化。 您可能需要手动重置这些通道。

4. 配置示例配置的通道。

   有关以下示例中使用的参数的更多信息，请参阅 第 43 页的『z/OS 的通道配置参数』。 这些示例用于连接 IBM MQ for z/OS 和 IBM MQ for Windows。 要在另一个平台上连接到 IBM MQ ， 请使用 第 43 页的『z/OS 的通道配置参数』 中相应表中的值，而不是 Windows 的值。

   a) 定义发送方通道，如以下示例中所示:

   对于 LU 6.2:

```
     Local Queue
            Object type :   QLOCAL
                   Name :   WINNT              F
                  Usage :   X (XmitQ)

     Remote Queue
            Object type :   QREMOTE
                   Name :   WINNT.REMOTEQ      D
  Name on remote system :   WINNT.LOCALQ       E
     Remote system name :   WINNT              C
     Transmission queue :   WINNT              F

     Sender Channel
          Channel name :   MVS.WINNT.SNA      G
       Transport type :   L (LU6.2)
Transmission queue name :   WINNT              F
        Connection name :   M3                13
```

   对于 TCP:

```
     Local Queue
            Object type :   QLOCAL
                   Name :   WINNT              F
                  Usage :   X (XmitQ)

     Remote Queue
            Object type :   QREMOTE
                   Name :   WINNT.REMOTEQ      D
```

```
   Name on remote system :   WINNT.LOCALQ       E
      Remote system name :   WINNT              C
      Transmission queue :   WINNT              F

      Sender Channel
            Channel name :   MVS.WINNT.TCP      H
          Transport type :   T (TCP)
 Transmission queue name :   WINNT              F
         Connection name :   winnt.tcpip.hostname
```

b) 定义接收方通道，如以下示例中所示:

对于 LU 6.2:

```
      Local Queue
             Object type :   QLOCAL
                    Name :   MVS.LOCALQ        B
                   Usage :   N (Normal)

      Receiver Channel
            Channel name :   WINNT.MVS.SNA     I
```

对于 TCP:

```
      Local Queue
             Object type :   QLOCAL
                    Name :   MVS.LOCALQ        B
                   Usage :   N (Normal)

      Receiver Channel
            Channel name :   WINNT.MVS.TCP     J
```

z/OS *z/OS* 的通道配置参数

为 z/OS 上的示例配置配置通道所需的参数。

步骤 第 42 页的『4』 (共 第 42 页的『在 IBM MQ for z/OS 上配置通道』 步) 描述了要在 z/OS 队列管理器上执行的配置，以实现 第 5 页的『示例: 为 IBM MQ 设置跨平台通信』中描述的通道。 第 42 页的『在 IBM MQ for z/OS 上配置通道』 中的示例用于连接 IBM MQ for z/OS 和 IBM MQ for Windows。 要在另一平台上连接到 IBM MQ，请使用相应表中的值来代替 Windows 的值。

注: 粗体 中的词是建议值，并反映在这些示例中使用的 IBM MQ 对象的名称。 您可以在产品安装中更改这些值，但如果更改了这些值，请确保在使用本部分中的示例时使用自己的值

本地节点的定义

表 *30:* 本地节点定义的配置示例

标识	参数名称	参考	使用的示例
A	队列管理器名称		**MVS**
B	本地队列名称		**MVS.LOCALQ**

连接到 Windows 上的 IBM MQ

Windows

表的此部分中的值必须与 第 39 页的『Windows 的通道配置参数』中使用的值匹配，如所指示。

表 *31:* 用于连接到 *Windows* 上的 *IBM MQ* 的配置示例

标识	参数名称	参考	使用的示例
C	远程队列管理器名称	A	**winnt**
D	远程队列名称		**WINNT.REMOTEQ**
E	远程系统上的队列名称	B	**WINNT.LOCALQ**

表 31: 用于连接到 Windows 上的 IBM MQ 的配置示例 (继续)			
标识	**参数名称**	**参考**	**使用的示例**
F	传输队列的名称		**winnt**
G	发送方 (LU 6.2) 通道名称		**MVS.WINNT.SNA**
H	发送方 (TCP) 通道名称		**MVS.WINNT.TCP**
I	接收方 (LU 6.2) 通道名称	G	**WINNT.MVS.SNA**
J	接收方 (TCP/IP) 通道名称	H	**WINNT.MVS.TCP**

连接到 AIX 上的 IBM MQ

AIX

表的此部分中的值必须与 第 11 页的『AIX 的通道配置参数』中使用的值匹配，如所指示。

表 32: 用于连接到 AIX 上的 IBM MQ 的配置示例			
标识	**参数名称**	**参考**	**使用的示例**
AIX AIX **与 *IBM MQ for AIX* 的连接**			
C	远程队列管理器名称	A	**AIX**
D	远程队列名称		**AIX。远程队列**
E	远程系统上的队列名称	B	**AIX.LOCALQ**
F	传输队列的名称		**AIX**
G	发送方 (LU 6.2) 通道名称		**MVS.AIX.SNA**
H	发送方 (TCP/IP) 通道名称		**MVS.AIX.TCP**
I	接收方 (LU 6.2) 通道名称	G	**AIX.MVS.SNA**
J	接收方 (TCP/IP) 通道名称	H	**AIX.MVS.TCP**

连接到 Linux 上的 IBM MQ

Linux

表的此部分中的值必须与 第 33 页的『Linux 的通道配置参数』中使用的值匹配，如所指示。

表 33: 用于连接到 Linux 上的 IBM MQ 的配置示例			
标识	**参数名称**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**Linux**
D	远程队列名称		**LINUX.REMOTEQ**
E	远程系统上的队列名称	B	**LINUX.LOCALQ**
F	传输队列的名称		**Linux**
G	发送方 (LU 6.2) 通道名称		**MVS.LINUX.SNA**
H	发送方 (TCP) 通道名称		**MVS.LINUX.TCP**
I	接收方 (LU 6.2) 通道名称	G	**LINUX.MVS.SNA**
J	接收方 (TCP/IP) 通道名称	H	**LINUX.MVS.TCP**

连接到 IBM i 上的 IBM MQ

IBM i

表的此部分中的值必须与 第 27 页的『IBM i 的通道配置参数』中使用的值匹配，如所指示。

表 *34:* 用于连接到 *IBM i* 上的 *IBM MQ* 的配置示例			
标识	**参数名**	**参考**	**使用的示例**
C	远程队列管理器名称	A	**AS400**
D	远程队列名称		**AS400.REMOTEQ**
E	远程系统上的队列名称	B	**AS400.LOCALQ**
F	传输队列的名称		**AS400**
G	发送方 (LU 6.2) 通道名称		**MVS.AS400.SNA**
H	发送方 (TCP/IP) 通道名称		**MVS.AS400.TCP**
I	接收方 (LU 6.2) 通道名称	G	**AS400.MVS.SNA**
J	接收方 (TCP/IP) 通道名称	H	**AS400.MVS.TCP**

z/OS Example: setting up IBM MQ cross-platform communication on z/OS using QSGs

This example shows how to set up communication links to a queue sharing group (QSG) from IBM MQ on Windows and AIX. You can also connect from z/OS to z/OS.

Before you begin

Setting up communication links from a queue sharing group to a platform other than z/OS is the same as described in “示例: 在 z/OS 上设置 IBM MQ 跨平台通信” on page 41.

For background information about this example and how to use it, see “示例: 为 IBM MQ 设置跨平台通信” on page 5 and “如何使用跨平台通信示例” on page 7.

Procedure

1. Establish a network connection using one of the following options.
   - Establish an LU 6.2 connection as described in “在队列共享组中建立 LU 6.2 连接” on page 45.
   - Establish a TCP connection using Sysplex Distributor as described in “Establishing a TCP connection using Sysplex Distributor” on page 49.
2. Define some channels to complete the configuration after the connection is established.
   See “在 IBM MQ for z/OS 上配置共享通道” on page 50 for details of this process.

z/OS 在队列共享组中建立 LU 6.2 连接

建立 LU 6.2 连接有两个步骤。 定义自己与网络的关系，并定义与合作伙伴的连接。

关于此任务

注: 此示例用于与 Windows 系统的连接，但对于其他平台，此任务相同。

过程

1. 使用 VTAM 通用资源具有一个连接名称以连接到队列共享组。

a) SYS1.PARMLIB(APPCPMxx) 包含 APPC 的启动参数。 您必须向此文件添加一行，以告知 APPC 在何处查找 sideinfo。

此行必须具有以下格式:

```
SIDEINFO
    DATASET(APPC.APPCSI)
```

b) 将另一行添加到 SYS1.PARMLIB(APPCPMxx)，用于定义要用于 IBM MQ LU 6.2 组侦听器的本地 LU 名。

您添加的行必须采用表单

```
LUADD ACBNAME(mvslu1)
      NOSCHED
      TPDATA(csq.appctp)
      GRNAME(mvsgr)
```

指定 ACBNAME (9)， TPDATA 和 GRNAME(10)的值。

NOSCHED 参数告诉 APPC，我们的新 LU 不使用 LU 6.2 调度程序 (ASCH)，而是有自己的一个。 TPDATA 是指 "事务程序" 数据集， LU 6.2 在其中存储有关事务程序的信息。 同样， IBM MQ 不使用此参数，但 LUADD 命令的语法需要此参数。

c) 使用以下命令启动 APPC 子系统:

```
START APPC,SUB=MSTR,APPC=xx
```

其中 *xx* 是在步骤 1 中添加了 LU 的 PARMLIB 成员的后缀。

注: 如果 APPC 已在运行，那么可以使用以下命令对其进行刷新:

```
SET APPC=xx
```

此操作的影响是累积的，即 APPC 不会丢失它对已在此成员或另一 PARMLIB 成员中定义的对象的了解。

d) 将新 LU 添加到合适的 VTAM 主节点定义。 这些通常在 SYS1.VTAMLST。

APL 定义将类似于所显示的样本。

```
        MVSLU APPL  ACBNAME=MVSLU1,     9
                    APPXC=YES,
                    AUTOSES=0,
                    DDRAINL=NALLOW,
                    DLOGMOD=#INTER,     6
                    DMINWML=10,
                    DMINWNR=10,
                    DRESPL=NALLOW,
                    DSESLIM=60,
                    LMDENT=19,
                    MODETAB=MTCICS,
                    PARSESS=YES,
                    VERIFY=NONE,
                    SECACPT=ALREADYV,
                    SRBEXIT=YES
```

e) 激活主节点。

您可以使用以下命令来执行此激活:

```
V,NET,ACT,majornode
```

f) 使用 APPC 实用程序 ATBSDFMU 将定义 LU 和通用资源名称的条目添加到 CPI-C 端信息数据集。

样本 JCL 位于 *thlqual*.SCSQPROC (CSQ4SIDE) 中 (其中 *thlqual* 是安装中 IBM MQ 数据集的目标库高级限定符。)

您添加的条目将类似于以下示例:

```
     SIADD
          DESTNAME(G1)           11
          MODENAME(#INTER)
          TPNAME(MQSERIES)
```

```
          PARTNER_LU(MVSLU1)     9
     SIADD
          DESTNAME(G2)           12
          MODENAME(#INTER)
          TPNAME(MQSERIES)
          PARTNER_LU(MVSGR)      10
```

g) 使用以下命令更改队列管理器对象以使用正确的分布式排队参数。
必须在队列管理器的 LUGROUP 属性中指定分配给队列管理器的本地 LU (9)。

```
ALTER QMGR LUGROUP(MVSLU1)
```

2. 通过向 CPI-C 端信息数据集添加条目来定义与合作伙伴的连接。
   a) 将条目添加到 CPI-C 端信息数据集以定义连接。
   用于执行此定义的样本 JCL 位于 *thlqual*.SCSQPROC (CSQ4SIDE) 中。
   您添加的条目如下所示:

```
     SIADD
          DESTNAME(M3)           13
          MODENAME(#INTER)       14
          TPNAME(MQSERIES)       15
          PARTNER_LU(WINNTLU)    16
```

下一步做什么

现在已建立连接。 您已准备好完成配置。

转至 第 50 页的『在 IBM MQ for z/OS 上配置共享通道』。

z/OS *Configuration parameters for an LU 6.2 connection*

The following table lists all the parameters required to set up communication from a z/OS system to IBM MQ on another platform.

The steps required to set up an LU 6.2 connection are described in “在队列共享组中建立 LU 6.2 连接” on page 45, with numbered cross-references to the parameters in the example.

Numbers in the Reference column indicate that the value must match that in the appropriate example elsewhere in this section. The examples that follow in this section refer to the values in the ID column. The entries in the Parameter Name column are explained in “Explanation of terms” on page 48.

Definition for local node using generic resources

Table 35. Configuration examples for the definition for the local node using generic resources

ID	Parameter name	Reference	Example used
1	Command prefix		**/cpf**
2	Network ID		**NETID**
3	Node name		**MVSPU**
6	Modename		**#INTER**
7	Local Transaction Program name		**MQSERIES**
8	LAN destination address		**400074511092**
9	Local LU name		**MVSLU1**
10	Generic resource name		**MVSGR**
11	Symbolic destination		**G1**
12	Symbolic destination for generic resource name		**G2**

Connection to IBM MQ on Windows

Windows

Table 36. Configuration examples for connecting to IBM MQ on Windows using LU 6.2			
ID	**Parameter name**	**Reference**	**Example used**
13	Symbolic destination		**M3**
14	Modename	21	**#INTER**
15	Remote Transaction Program name	7	**MQSERIES**
16	Partner LU name	5	**WINNTLU**
21	Remote node ID	4	**05D 30F65**

Connection to IBM MQ on AIX

AIX

Table 37. Configuration examples for connecting to IBM MQ on AIX using LU 6.2			
ID	**Parameter name**	**Reference**	**Example used**
13	Symbolic Destination		**M4**
14	Modename	18	**#INTER**
15	Remote Transaction Program name	6	**MQSERIES**
16	Partner LU name	4	**AIXLU**

Explanation of terms

1 Command prefix
This term is the unique command prefix of your IBM MQ for z/OS queue manager subsystem. The z/OS system programmer defines this value at installation time, in SYS1.PARMLIB(IEFSSNss), and can tell you the value.

2 Network ID
The VTAM startup procedure in your installation is partly customized by the ATCSTRxx member of the data set referenced by the DDNAME VTAMLST. The Network ID is the value specified for the NETID parameter in this member. For Network ID, you must specify the name of the NETID that owns the IBM MQ communications subsystem. Your network administrator can tell you the value.

3 Node name
VTAM, being a low-entry network node, does not have a Control Point name for Advanced Peer-to-Peer Networking (APPN) use. It does however have a system services control point name (SSCPNAME). For node name, you must specify the name of the SSCP that owns the IBM MQ communications subsystem. This value is defined in the same ATCSTRxx member as the Network ID. Your network administrator can tell you the value.

9 Local LU name
A logical unit (LU) is software that serves as an interface or translator between a transaction program and the network. It manages the exchange of data between transaction programs. The local LU name is the unique VTAM APPLID of this IBM MQ subsystem. Your network administrator can tell you this value.

11 12 13 Symbolic destination
This term is the name you give to the CPI-C side information profile. You need a side information entry for each LU 6.2 listener.

6 14 Modename
This term is the name given to the set of parameters that control the LU 6.2 conversation. An entry with this name and similar attributes must be defined at each end of the session. In VTAM, this corresponds to a mode table entry. You network administrator can assign this table entry to you.

7 15 Transaction Program name
IBM MQ applications trying to converse with this queue manager specify a symbolic name for the program to be run at the receiving end. This has been specified in the TPNAME attribute on the channel definition at the sender. For simplicity, wherever possible use a transaction program name of MQSERIES, or in the case of a connection to VSE/ESA, where the length is limited to 4 bytes, use MQTP.

See Defining an LU6.2 connection for z/OS using APPC/MVS for more information.

8 LAN destination address
This term is the LAN destination address that your partner nodes use to communicate with this host. When you are using a 3745 network controller, it is the value specified in the LOCADD parameter for the line definition to which your partner is physically connected. If your partner nodes use other devices such as 317X or 6611 devices, the address is set during the customization of those devices. Your network administrator can tell you this value.

10 Generic resource name
A generic resource name is a unique name assigned to a group of LU names used by the channel initiators in a queue sharing group.

16 Partner LU name
This term is the LU name of the IBM MQ queue manager on the system with which you are setting up communication. This value is specified in the side information entry for the remote partner.

21 Remote node ID
For a connection to Windows, this ID is the ID of the local node on the Windows system with which you are setting up communication.

z/OS Establishing a TCP connection using Sysplex Distributor

You can set up Sysplex distributor to use one connection name to connect to the queue sharing group.

Procedure

1. Define a Distributed DVIPA address as follows:
   a) Add a DYNAMICXCF statement to the IPCONFIG. This statement is used for inter-image connectivity using dynamically created XCF TCP/IP links.
   b) Use the VIPADYNAMIC block on each image in the Sysplex.

      On the owning image, code a VIPADEFINE statement to create the DVIPA Then code a VIPADISTRIBUTE statement to distribute it to all other or selected images.

      On the backup image, code a VIPABACKUP statement for the DVIPA address.
2. Add the SHAREPORT option for the port to be shared in the PORT reservation list in the PROFILE data set if more than one channel initiator is to be started on any LPAR in the sysplex.

   See PORT statement in the *z/OS Communications Server: IP Configuration Reference* for more information.

   When you have completed these steps, the TCP connection is established. You are ready to complete the configuration.

What to do next

Go to “在 IBM MQ for z/OS 上配置共享通道” on page 50.

z/OS 在 IBM MQ for z/OS 上配置共享通道

通过启动通道启动程序并针对配置发出相应命令来配置共享通道。

关于此任务

一次只能有一个共享通道实例在运行。 如果尝试启动通道的第二个实例，那么它将失败 (错误消息因其他因素而异)。 共享同步队列将跟踪通道状态。

要点: 如果通道协商检测到消息序号在每一端都不同，那么 IBM MQ 通道不会成功初始化。 您可能需要手动重置此值。

过程

1. 使用以下命令启动通道启动程序:

```
/cpf   START CHINIT
```

2. 使用以下命令启动 LU6.2 组侦听器:

```
/cpf   START LSTR TRPTYPE(LU62) LUNAME( G1 ) INDISP(GROUP)
```

   G1 的 LUNAME 是指您为 LU 指定的符号名称 (11)。

3. 如果您使用的是使用综合系统分发器的虚拟 IP 寻址，并且想要侦听特定地址，请使用以下命令:

```
/cpf   START LSTR TRPTYPE(TCP) PORT(1555) IPADDR( mvsvipa ) INDISP(GROUP)
```

4. 配置示例配置的通道。

   有关以下示例中使用的参数的更多信息，请参阅 第 51 页的『Shared channel configuration parameters』。 这些示例用于连接 IBM MQ for z/OS 和 Windows。 要在另一个平台上连接到 IBM MQ，请使用 第 51 页的『Shared channel configuration parameters』 中表中的相应值，而不是 Windows 的值。

   a) 定义共享发送方通道，如以下示例中所示。

   使用 LU 6.2:

```
Local Queue
            Object type :   QLOCAL
                   Name :   WINNT              F
                  Usage :   X (XmitQ)
            Disposition :   SHARED

     Remote Queue
            Object type :   QREMOTE
                   Name :   WINNT.REMOTEQ      D
  Name on remote system :   WINNT.LOCALQ       E
     Remote system name :   WINNT              C
     Transmission queue :   WINNT              F
            Disposition :   GROUP

     Sender Channel
          Channel name :    MVS.WINNT.SNA      G
       Transport type :     L (LU6.2)
Transmission queue name :   WINNT              F
        Connection name :   M3                13
            Disposition :   GROUP
```

   使用 TCP

```
     Local Queue
            Object type :   QLOCAL
                   Name :   WINNT              F
                  Usage :   X (XmitQ)
            Disposition :   SHARED

     Remote Queue
            Object type :   QREMOTE
                   Name :   WINNT.REMOTEQ      D
```

```
   Name on remote system :   WINNT.LOCALQ       E
      Remote system name :   WINNT              C
      Transmission queue :   WINNT              F
             Disposition :   GROUP

      Sender Channel
            Channel name :   QSG.WINNT.TCP      H
          Transport type :   T (TCP)
Transmission queue name :   WINNT              F
          Connection name :   winnt.tcpip.hostname
             Disposition :   GROUP
```

b) 定义共享接收方通道，如以下示例中所示。

使用 LU 6.2:

```
      Local Queue
             Object type :   QLOCAL
                    Name :   QSG.SHAREDQ        B
                   Usage :   N (Normal)
             Disposition :   SHARED

      Receiver Channel
            Channel name :   WINNT.QSG.SNA      I
             Disposition :   GROUP
```

使用 TCP:

```
      Local Queue
             Object type :   QLOCAL
                    Name :   QSG.SHAREDQ        B
                   Usage :   N (Normal)
             Disposition :   SHARED

      Receiver Channel
            Channel name :   WINNT.QSG.TCP      J
             Disposition :   GROUP
```

z/OS *Shared channel configuration parameters*

The parameters needed to configure a shared channel for the example configuration on z/OS.

Step “4” on page 50 of “在 IBM MQ for z/OS 上配置共享通道” on page 50 describes the configuration to be performed on the z/OS queue manager to implement the channel described in “示例: 为 IBM MQ 设置跨平台通信” on page 5. The examples in “在 IBM MQ for z/OS 上配置共享通道” on page 50 are for connecting IBM MQ for z/OS and Windows. To connect to IBM MQ on another platform, use the values from the appropriate table in place of the values for Windows.

Note: The words in **bold** are suggested values and reflect the names of IBM MQ objects used throughout these examples. You can change them in your product installation but, if you do, make sure that you use your own values when working through the examples in this section.

Definition for local node

Table 38. Configuration examples for the definition for the local node

ID	Parameter Name	Reference	Example Used
A	Queue Manager Name		**QSG**
B	Local queue name		**QSG.SHAREDQ**

Connection to IBM MQ on Windows

Windows

The values in this section of the table must match the values used in “Windows 的通道配置参数” on page 39, as indicated.

Table 39. Configuration examples for connecting to to IBM MQ on Windows			
ID	**Parameter name**	**Reference**	**Example used**
C	Remote queue manager name	A	**WINNT**
D	Remote queue name		**WINNT.REMOTEQ**
E	Queue name at remote system	B	**WINNT.LOCALQ**
F	Transmission queue name		**WINNT**
G	Sender (LU 6.2) channel name		**QSG.WINNT.SNA**
H	Sender (TCP) channel name		**QSG.WINNT.TCP**
I	Receiver (LU 6.2) channel name	G	**WINNT.QSG.SNA**
J	Receiver (TCP/IP) channel name	H	**WINNT.QSG.TCP**

Connection to IBM MQ on AIX

AIX

The values in this section of the table must match the values used in “AIX 的通道配置参数” on page 11, as indicated.

Table 40. Configuration examples for connecting to IBM MQ on AIX			
ID	**Parameter name**	**Reference**	**Example used**
C	Remote queue manager name		**AIX**
D	Remote queue name		**AIX.REMOTEQ**
E	Queue name at remote system	B	**AIX.LOCALQ**
F	Transmission queue name		**AIX**
G	Sender (LU 6.2) channel name		**QSG.AIX.SNA**
H	Sender (TCP/IP) channel name		**QSG.AIX.TCP**
I	Receiver (LU 6.2) channel name	G	**AIX.QSG.SNA**
J	Receiver (TCP/IP) channel name	H	**AIX.QSG.TCP**

z/OS Example: setting up IBM MQ cross-platform communication for intra-group queuing on z/OS

This example shows how a typical payroll query application that currently uses distributed queuing to transfer small messages between queue managers could be migrated to use queue sharing groups and shared queues.

About this task

Three configurations are described to illustrate the use of distributed queuing, intra-group queuing with shared queues, and shared queues. The associated diagrams show only the flow of data in one direction, that is, from queue manager QMG1 to queue manager QMG3.

Procedure

1. Set up and run Configuration 1.
   For more information, see “设置和运行配置 1” on page 53.

2. Set up and run Configuration 2.
   For more information, see “设置和运行配置 2” on page 55.
3. Set up and run Configuration 3.
   For more information, see “设置和运行配置 3” on page 57.

What to do next

You can expand the example in a number of ways by:

- Using channel triggering as well as application (PAYROLL and PAYROLL.REPLY queue) triggering.
- Configuring for communication using LU6.2.
- Configuring more queue managers to the queue sharing group. Then the server application can be cloned to run on other queue manager instances to provide multiple servers for the PAYROLL query queue.
- Increasing the number of instances of the payroll query requesting application to demonstrate the processing of requests from multiple clients.
- Using security (IGQAUT and IGQUSER).

z/OS 设置和运行配置 1

配置 1 描述了当前如何使用分布式排队在队列管理器 QMG1 和 QMG3 之间传输消息。

关于此任务

配置 1 显示了一个分布式排队系统，该系统用于将队列管理器 QMG1 接收到的消息从工资单查询传输到队列管理器 QMG2，然后最终传输到队列管理器 QMG3，以发送到工资单服务器。

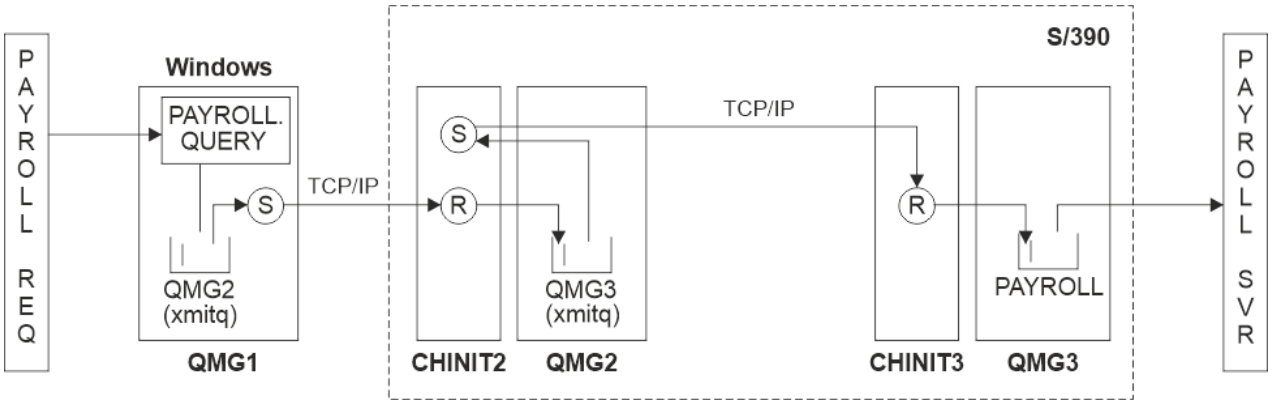

图 *2:* 配置 *1: z/OS* 使用组内排队

操作流程如下所示:

1. 使用连接到队列管理器 QMG1 的工资单请求应用程序输入查询。
2. 工资单请求应用程序将查询放到远程队列 PAYROLL.QUERY。 作为队列 PAYROLL.QUERY 解析为传输队列 QMG2，将查询放入传输队列 QMG2。
3. 队列管理器 QMG1 上的发送方通道 (S) 将查询传递到队列管理器 QMG2 上的伙伴接收方通道 (R)。
4. 队列管理器 QMG2 上的接收方通道 (R) 将查询置于队列管理器 QMG3 上的队列 PAYROLL。 由于 QMG3 上的队列 PAYROLL 解析为传输队列 QMG3，因此会将查询放入传输队列 QMG3。
5. 队列管理器 QMG2 上的发送方通道 (S) 将查询传递到队列管理器 QMG3 上的伙伴接收方通道 (R)。
6. 队列管理器 QMG3 上的接收方通道 (R) 将查询放入本地队列 PAYROLL。
7. 连接到队列管理器 QMG3 的工资单服务器应用程序从本地队列 PAYROLL 检索查询，处理该查询并生成适当的应答。

配置 1 所需的定义如下所示 (请注意，这些定义不考虑触发，并且仅提供用于使用 TCP/IP 进行通信的通道定义)。

过程

1. QMG1:
   a) 设置远程队列定义:

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QMG3') REPLACE +
PUT(ENABLED) RNAME(PAYROLL) RQMNAME(QMG3) XMITQ(QMG2)
```

   b) 设置传输队列定义:

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

   c) 使用 TCP/IP 设置发送方通道定义:

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +
DESCR('Sender channel to QMG2') XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

   注: 将 MVSQMG2(1415) 替换为队列管理器连接名称和端口。

   d) 使用 TCP/IP 设置接收方通道定义:

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG2')
```

   e) 设置应答队列定义:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to payroll queries sent to QMG3')
```

2. QMG2:
   a) 设置传输队列定义:

```
DEFINE QLOCAL(QMG1) DESCR('Transmission queue to QMG1') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)

DEFINE QLOCAL(QMG3) DESCR('Transmission queue to QMG3') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

   b) 使用 TCP/IP 设置发送方通道定义:

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +
DESCR('Sender channel to QMG1') XMITQ(QMG1) CONNAME('WINTQMG1(1414)')
```

   注: 将 WINTQMG1(1414) 替换为队列管理器连接名称和端口。

```
DEFINE CHANNEL(QMG2.TO.QMG3) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +
DESCR('Sender channel to QMG3') XMITQ(QMG3) CONNAME('MVSQMG3(1416)')
```

   注: 将 MVSQMG3(1416) 替换为队列管理器连接名称和端口。

   c) 使用 TCP/IP 设置接收方通道定义:

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG1')

DEFINE CHANNEL(QMG3.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG3')
```

3. QMG3:
   a) 设置本地队列定义:

```
DEFINE QLOCAL(PAYROLL) DESCR('Payroll query request queue') REPLACE +
PUT(ENABLED) USAGE(NORMAL) GET(ENABLED) SHARE
```

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

b) 使用 TCP/IP 设置发送方通道定义:

```
DEFINE CHANNEL(QMG3.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +
DESCR('Sender channel to QMG2) XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

注: 将 MVSQMG2(1415) 替换为队列管理器连接名称和端口。

c) 使用 TCP/IP 设置接收方通道定义:

```
DEFINE CHANNEL(QMG2.TO.QMG3) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG2)
```

4. 使用以下过程来运行配置 1:
   a) 启动队列管理器 QMG1， QMG2 和 QMG3。
   b) 启动 QMG2 和 QMG3 的通道启动程序。
   c) 启动 QMG1 上的侦听器以侦听端口 1414 ，启动 QMG2 以侦听端口 1415 ，启动 QMG3 以侦听端口 1416。
   d) 在 QMG1， QMG2 和 QMG3 上启动发送方通道。
   e) 启动工资单查询，请求连接到 QMG1 的应用程序。
   f) 启动连接到 QMG3 的工资单服务器应用程序。
   g) 向 QMG3 提交薪资查询请求，并等待薪资回复。

z/OS 设置和运行配置 2

配置 2 描述了如何使用队列共享组和组内排队 (对后端薪资服务器应用程序没有任何影响) 在队列管理器 QMG1 和 QMG3 之间传输消息。

关于此任务

配置 2 显示了一个分布式排队系统，该系统使用队列共享组和组内排队将消息从工资单请求应用程序传输到工资单服务器。 此配置将不需要队列管理器 QMG2 和 QMG3 之间的通道定义，因为组内排队用于在这两个队列管理器之间传输消息。

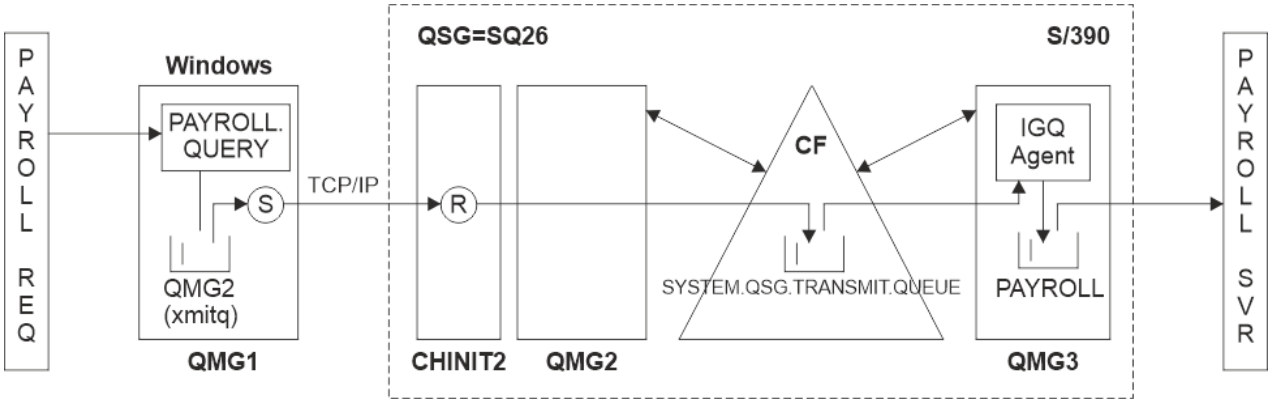

图 *3:* 配置 *2*

操作流程如下所示:

1. 使用连接到队列管理器 QMG1 的工资单请求应用程序输入查询。
2. 工资单请求应用程序将查询放到远程队列 PAYROLL.QUERY。 作为队列 PAYROLL.QUERY 解析为传输队列 QMG2，将查询放入传输队列 QMG2。
3. 队列管理器 QMG1 上的发送方通道 (S) 将查询传递到队列管理器 QMG2 上的伙伴接收方通道 (R)。

4. 队列管理器 QMG2 上的接收方通道 (R) 将查询置于队列管理器 QMG3 上的队列 PAYROLL。 由于 QMG3 上的队列 PAYROLL 解析为共享传输队列 SYSTEM.QSG.TRANSMIT.QUEUE，查询将放入共享传输队列 SYSTEM.QSG.TRANSMIT.QUEUE。
5. 队列管理器 QMG3 上的 IGQ 代理程序从共享传输队列 SYSTEM.QSG.TRANSMIT.QUEUE，并将其放在队列管理器 QMG3 上的本地队列 PAYROLL 上。
6. 连接到队列管理器 QMG3 的工资单服务器应用程序从本地队列 PAYROLL 检索查询，处理该查询并生成适当的应答。

注意:

- 薪资查询示例仅传输小消息。 如果需要同时传输持久消息和非持久消息，那么可以建立 "配置 1" 和 "配置 2" 的组合，以便可以使用分布式排队路径来传输大型消息，而可以使用可能更快的组内排队路径来传输小型消息。
- 这些定义不考虑触发，并且仅提供用于使用 TCP/IP 进行通信的通道定义。
- 此示例假定您已将队列管理器 QMG2 和 QMG3 配置为同一队列共享组的成员。

过程

1. QMG1:
   a) 设置远程队列定义:

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QMG3') REPLACE +
PUT(ENABLED) RNAME(PAYROLL) RQMNAME(QMG3) XMITQ(QMG2)
```

   b) 设置传输队列定义:

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

   c) 设置 TCP/IP 的发送方通道定义:

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +
DESCR('Sender channel to QMG2') XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

   注: 将 MVSQMG2(1415) 替换为队列管理器连接名称和端口。

   d) 设置 TCP/IP 的接收方通道定义:

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG2')
```

   e) 设置应答队列定义:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to payroll queries sent to QMG3')
```

2. QMG2:
   a) 设置传输队列定义:

```
DEFINE QLOCAL(QMG1) DESCR('Transmission queue to QMG1') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)

DEFINE QLOCAL(SYSTEM.QSG.TRANSMIT.QUEUE) QSGDISP(SHARED) +
DESCR('IGQ Transmission queue') REPLACE PUT(ENABLED) USAGE(XMITQ) +
GET(ENABLED) INDXTYPE(CORRELID) CFSTRUCT('APPLICATION1') +
DEFSOPT(SHARED) DEFPSIST(NO)
```

   注: 将 APPLICATION1 替换为定义的 CF 结构名称。 此外，此队列是共享队列，只需在队列共享组中的其中一个队列管理器上定义。

   b) 设置 TCP/IP 的发送方通道定义:

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +
DESCR('Sender channel to QMG1') XMITQ(QMG1) CONNAME('WINTQMG1(1414)')
```

注: 将 WINTQMG1(1414) 替换为队列管理器连接名称和端口。

c) 设置 TCP/IP 的接收方通道定义:

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG1')
```

d) 设置队列管理器定义:

```
ALTER QMGR IGQ(ENABLED)
```

3. QMG3:

a) 设置本地队列定义:

```
DEFINE QLOCAL(PAYROLL) DESCR('Payroll query request queue') REPLACE +
PUT(ENABLED) USAGE(NORMAL) GET(ENABLED) SHARE
```

b) 设置队列管理器定义:

```
ALTER QMGR IGQ(ENABLED)
```

4. 使用以下过程来运行配置 2:

a) 启动队列管理器 QMG1， QMG2 和 QMG3。

b) 启动 QMG2 的通道启动程序。

c) 启动 QMG1 上的侦听器以侦听端口 1414 ，启动 QMG2 以侦听端口 1415。

d) 在 QMG1 和 QMG2 上启动发送方通道。

e) 启动工资单查询，请求连接到 QMG1 的应用程序。

f) 启动连接到 QMG3 的工资单服务器应用程序。

g) 向 QMG3 提交薪资查询请求，并等待薪资回复。

z/OS 设置和运行配置 3

配置 3 描述了如何使用队列共享组和共享队列 (对后端薪资服务器应用程序没有任何影响) 在队列管理器 QMG1 和 QMG3 之间传输消息。

关于此任务

配置 3 显示了一个分布式排队系统，该系统使用队列共享组和共享队列在队列管理器 QMG1 和队列管理器 QMG3 之间传输消息。

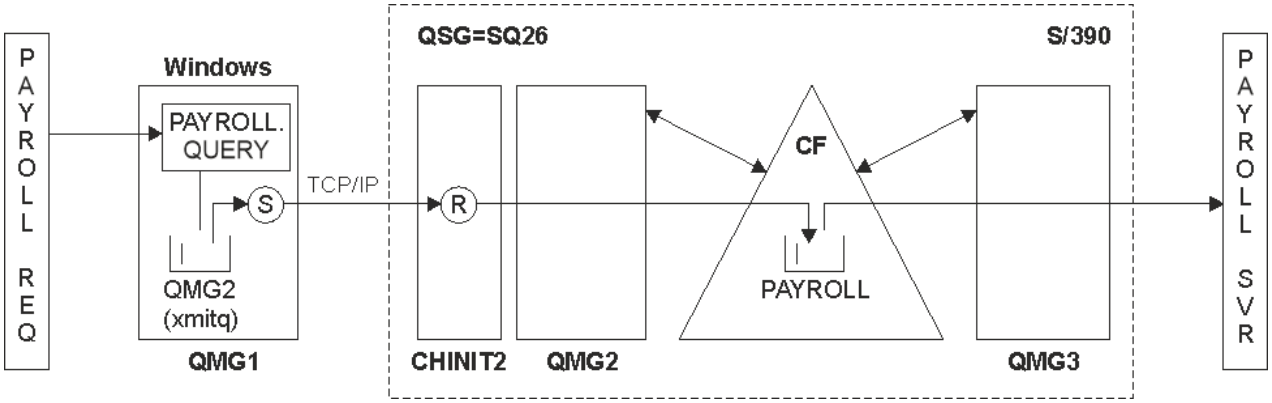

图 *4:* 配置 *3*

操作流为:

1. 使用连接到队列管理器 QMG1 的工资单请求应用程序输入查询。
2. 工资单请求应用程序将查询放到远程队列 PAYROLL.QUERY。 作为队列 PAYROLL.QUERY 解析为传输队列 QMG2，将查询放入传输队列 QMG2。

3. 队列管理器 QMG1 上的发送方通道 (S) 将查询传递到队列管理器 QMG2 上的伙伴接收方通道 (R)。
4. 队列管理器 QMG2 上的接收方通道 (R) 将查询放入共享队列 PAYROLL。
5. 连接到队列管理器 QMG3 的工资单服务器应用程序从共享队列 PAYROLL 检索查询，处理该查询并生成适当的应答。

此配置当然是最简单的配置。 但是，您需要配置分布式排队或组内排队，以将应答 (由连接到队列管理器 QMG3 的工资单服务器应用程序生成) 从队列管理器 QMG3 传输到队列管理器 QMG2，然后再传输到队列管理器 QMG1。

有关用于将回复传输回工资单请求应用程序的配置，请参阅 第 141 页的『Example: planning a message channel for z/OS using queue sharing groups』。

注意:

- 仅提供用于使用 TCP/IP 进行通信的通道定义。
- 此示例假定您已将队列管理器 QMG2 和 QMG3 配置为同一队列共享组的成员。
- QMG3 上不需要任何定义。

过程

1. QMG1:
   a) 设置远程队列定义:

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QMG3') REPLACE +
PUT(ENABLED) RNAME(PAYROLL) RQMNAME(QMG3) XMITQ(QMG2)
```

   b) 设置传输队列定义:

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

   c) 设置发送方通道定义:

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QMG2') XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

   注: 将 MVSQMG2(1415) 替换为队列管理器连接名称和端口。

   d) 设置传输通道定义:

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG2')
```

   e) 设置应答队列定义:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to payroll queries sent to QMG3')
```

2. QMG2:
   a) 设置传输队列定义:

```
DEFINE QLOCAL(QMG1) DESCR('Transmission queue to QMG1') REPLACE +
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

   b) 设置 TCP/IP 的发送方通道定义:

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QMG1') XMITQ(QMG1) CONNAME('WINTQMG1(1414)')
```

   注: 将 WINTQMG1(1414) 替换为队列管理器连接名称和端口。

   c) 设置 TCP/IP 的接收方通道定义:

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QMG1')
```

d) 设置本地队列定义:

```
DEFINE QLOCAL(PAYROLL) QSGDISP(SHARED) DESCR('Payroll query request queue') +
REPLACE PUT(ENABLED) USAGE(NORMAL) GET(ENABLED) SHARE +
DEFSOPT(SHARED) DEFPSIST(NO) CFSTRUCT(APPLICATION1)
```

注: 将 APPLICATION1 替换为定义的 CF 结构名称。 同样，作为共享队列的此队列只需要在队列共享组中的其中一个队列管理器上定义。

3. 使用以下过程来运行配置 3:
   a) 启动队列管理器 QMG1， QMG2 和 QMG3。
   b) 启动 QMG2 的通道启动程序。
   c) 启动 QMG1 上的侦听器以侦听端口 1414 ，启动 QMG2 以侦听端口 1415。
   d) 在 QMG1 和 QMG2 上启动发送方通道。
   e) 启动工资单查询，请求连接到 QMG1 的应用程序。
   f) 启动连接到 QMG3 的工资单服务器应用程序。
   g) 向 QMG3 提交薪资查询请求，并等待薪资回复。

Linux IBM i AIX IBM MQ 文件系统许可权已应用于 /var/mqm

以下信息描述了应用于 /var/mqm/ 下的文件和目录的安全性以及设置文件系统许可权的原因。 为了确保 IBM MQ 的正确操作，不应更改 IBM MQ 设置的文件系统许可权

crtmqdir 命令

如果您的企业由于任何原因更改了任何 /var/mqm 文件许可权，那么可以使用 **crtmqdir** 命令来更新许可权或添加目录

AIX， Linux 和 IBM i 上的 IBM MQ 文件系统安全性

IBM MQ 数据目录 (/var/mqm) 下的文件用于存储:

- IBM MQ 配置数据
- 应用程序数据 (IBM MQ 对象和 IBM MQ 消息中包含的数据)
- 运行时控制信息
- 监视信息 (消息和 FFST 文件)

使用文件系统许可权来控制对此数据的访问，其中部分数据可供所有用户访问，而其他数据仅限于 IBM MQ 管理员组 "mqm" (或 IBM i 上的 QMQM)的成员。

在以下三个类别中授予访问权:

仅 mqm 组

此类别中的文件和目录仅可供 IBM MQ 管理员 ("mqm" 组的成员) 和 IBM MQ 队列管理器进程访问。

这些文件和目录的文件许可权为:

```
    -rwxrwx---    mqm:mqm      (UNIX and Linux)
    -rwxrwx---    QMQMADM:QMQM (IBM i)
```

此类别中的文件和目录的示例为:

```
    /var/mqm/qmgrs/QMGR/qm.ini
    /var/mqm/qmgrs/QMGR/channel/
    /var/mqm/qmgrs/QMGR/channel/SYSTEM!DEF!SCRVONN
    /var/mqm/qmgrs/QMGR/queues/
    /var/mqm/qmgrs/QMGR/queues/SYSTEM!DEFAULT!LOCAL!QUEUES/
    /var/mqm/qmgrs/QMGR/errors/
    /var/mqm/qmgrs/QMGR/errors/AMQERR01.LOG
    /var/mqm/qmgrs/QMGR/ssl/
```

```
/var/mqm/qmgrs/QMGR/@qmgr/
/var/mqm/qmgrs/QMGR/@qmpersist/
...
```

所有用户读访问权-mqm 组成员读和写访问权

此类别中的文件和目录可以由所有用户读取，但只有 "mqm" 组的成员才能修改这些文件并处理这些目录。

这些文件和目录的文件许可权为:

```
-rwxrwxr-x    mqm:mqm      (UNIX and Linux)
-rwxrwxr-x    QMQMADM:QMQM (IBM i)
```

此类别中的文件和目录的示例为:

```
/var/mqm/mqs.ini
/var/mqm/exits/
/var/mqm/qmgrs/
/var/mqm/qmgrs/QMGR/
/var/mqm/qmgrs/QMGR/@app/
/var/mqm/qmgrs/QMGR/@ipcc/
```

注意: 您只应该设置对可执行文件和脚本的执行许可权。 例如，在 Linux 上，当 **crtmqm** 命令运行时，将设置以下文件许可权:

```
-rw-rw---- mqm mqm /var/mqm/qmgrs/QMGR/qm.ini
-rw-rw---- mqm mqm /var/mqm/qmgrs/QMGR/channel/SYSTEM!DEF!SCRVONN
-rw-rw---- mqm mqm /var/mqm/qmgrs/QMGR/errors/AMQERR01.LOG
-rw-rw-r-- mqm mqm /var/mqm/mqs.ini
```

IBM MQ 8.0:

```
/var/mqm/sockets/@SYSTEM
/var/mqm/sockets/QMGR/@app/hostname
/var/mqm/sockets/QMGR/@ipcc/hostname
```

所有用户读写访问权

对所有用户具有读写访问权的文件

IBM MQ 没有具有全局可写文件许可权 (777) 的 常规 文件。 但是，有许多 特殊 文件显示为具有全局可写文件许可权。

这些特殊文件没有提供安全漏洞。 虽然许可权显示为 777，但它们不是 常规 文件，您无法直接写入这些文件。

这些特殊文件包括:

符号链接

符号链接在其许可权开始时由 "l" 字符标识。 对符号链接的许可权对谁能够访问目标文件没有影响，因为对命令的访问由对符号链接的目标的许可权控制。

在大多数 AIX and Linux 系统上，无法更改对符号链接的许可权，因此它们始终显示为 lrwxrwxrwx。

套接字文件

套接字文件是由于创建 UNIX 域套接字的过程而由操作系统创建的特殊文件。 这些文件可以由 "s" 在文件许可权开始时标识，即 srwxrwxrwx。

对该文件的许可权不授予对该文件本身的访问权，而是定义谁可以连接到 UNIX 域套接字。

IBM MQ 使用许多这些套接字文件，并且始终根据允许与套接字通信的人员来设置许可权。

以下目录包含对所有用户 (srwxrwxrwx) 具有读/写许可权的套接字文件。

IBM MQ 8.0:

```
/var/mqm/sockets/QMGR/zsocketEC/hostname/Zsocket_*
```

由使用隔离绑定连接到 IBM MQ 的应用程序使用的套接字文件。

```
/var/mqm/sockets/QMGR/@ipcc/ssem/hostname/*
```

对所有用户具有读写访问权的目录

有时 IBM MQ 应用程序需要在 IBM MQ 数据目录下创建文件。 为了确保应用程序能够在需要时创建文件，向许多目录授予了全局写访问权，这意味着系统上的任何用户都可以在该目录中创建文件。

除了错误日志文件 (可由 "mqm" 组的任何成员写入) 之外，在这些目录中创建的所有文件都具有受限许可权 (仅允许文件创建者写访问权)。 这允许系统管理员跟踪写入这些目录中文件的所有数据的用户标识。

/var/mqm/errors/

此目录包含系统错误日志文件和 FFST 文件。 此目录的许可权为 "drwxrwsrwt"，这意味着系统上的所有用户都可以在此目录中创建文件。

SetGroupId bit "s" 指示在此目录中创建的所有文件都具有组所有权 "mqm"。

缺省情况下，此目录上未设置 "t" 粘性位，但 IBM MQ 管理员可以显式设置此值，以允许用户仅删除他们创建的文件。

注: IBM i 此功能在 IBM i 上不可用。

AMQERR0*.LOG

这些错误日志文件只能由 group 的成员直接写入，但任何用户都可以读取写入这些文件的消息 (许可权: -rw-rw-r--)。

AMQnnnnn. * .FDC

这些文件包含在队列管理器或用户写入的应用程序中发生错误时写入的 FFST 信息。 这些文件是使用许可权 -rw-r-----创建的。

/var/mqm/trace/

当启用 IBM MQ 跟踪时，会将跟踪文件写入此目录。 IBM MQ 跟踪由与启用了跟踪的队列管理器关联的所有进程写入。

此目录的许可权为 "drwxrwsrwt"，这意味着系统上的所有用户都可以在此目录中创建文件。

SetGroupId bit "s" 指示在此目录中创建的所有文件都具有 'mqm' 的组所有权。

缺省情况下，此目录上未设置 "t" 粘性位，但 IBM MQ 管理员可以显式设置此值，以允许用户仅删除他们创建的文件。

注: IBM i 此功能在 IBM i 上不可用。

AMQnnnnn. * .TRC

这些文件包含正在跟踪的每个进程所写入的跟踪数据，并且是使用许可权 -rw-r----- 创建的

此目录上的许可权为 drwxrwsrwt，在此目录中创建的套接字文件的许可权为 srwx------。

IBM MQ 8.0:

```
/var/mqm/sockets/QMGR/zsocketapp/hostname/
```

此目录由使用 隔离 绑定连接到 IBM MQ 队列管理器的应用程序使用。 在连接处理期间，套接字文件由此目录中的连接应用程序创建。 在与队列管理器建立连接后，将除去套接字文件。

此目录上的许可权为 drwxrwsrwt，在此目录中创建的套接字文件的许可权为 srwx------。

此目录上的 SetGroupId bit 's' 可确保在此目录中创建的所有文件都具有 'mqm' 的组所有权。

在所有平台 (IBM i 除外) 上，此目录还设置了 "t" 粘性位，这将阻止用户删除除其所有者以外的任何文件。 这将阻止未经授权的用户删除他们不拥有的文件。

```
/var/mqm/sockets/QMGR/@ipcc/ssem/hostname/
/var/mqm/sockets/QMGR/@app/ssem/hostname/
```

AIX 对于使用 共享 绑定连接到 IBM MQ 的进程，可以使用 UNIX 域套接字在应用程序与队列管理器之间进行同步。 当使用 UNIX 域套接字时，将在这些目录中创建相关联的套接字文件。

这些目录上的许可权为 drwxrwsrwt，在这些目录中创建的套接字文件的许可权为 srwxrwxrwx。

这些目录上的 SetGroupId 位 "s" 可确保在这些目录中创建的所有文件都具有组所有权 "mqm"。

在 除 IBM i 之外的所有平台上，这些目录还设置了 "t" 粘滞位，这将阻止用户删除除其所有者以外的任何文件。 这将阻止未经授权的用户删除他们不拥有的文件。

主

使用未注册或未安装的 IBM MQ 版本 (例如，可重新分发的客户机) 时，将创建 ${HOME}/.mqm 目录。

将创建该目录，以便 IBM MQ 能够使用符合 **sun_path** 长度的路径来访问其套接字文件。 如果 IBM MQ 无法写入 HOME 目录，那么您将收到错误消息。

IBM MQ 使用 System V IPC 资源

IBM MQ 使用 System V 共享内存和信号量进行进程间通信。 这些资源根据其用于具有相应所有权和访问许可权的每个组的方式进行分组。

要验证系统上的哪些 System V IPC 资源属于 IBM MQ，可以执行以下操作:

- 检查所有权。

  拥有 IBM MQ System V IPC 资源的用户始终是 AIX and Linux 平台上的 "mqm" 用户。 在 IBM i 上，拥有用户为 "QMQM"。

- IBM MQ 8.0 和更高版本，使用 amqspdbg 实用程序。

  IBM MQ 随附的 amqspdbg 实用程序可用于显示给定队列管理器的共享内存和信号量标识。

  必须对 IBM MQ 创建的 System V 资源的 "system" 组发出一次命令

```
# amqspbg -z -I
```

  然后对系统上的每个队列管理器进行四次，以获取 IBM MQ 所使用的 System V 资源的完整列表。 在以下示例中，假定队列管理器名称为 QMGR1：

```
# amqspdbg -i QMGR1 -I
# amqspdbg -q QMGR1 -I
# amqspdbg -p QMGR1 -I
# amqspdbg -a QMGR1 -I
```

IBM MQ 创建的 System V 资源上的访问许可权设置为仅向允许的用户授予正确级别的访问权。 IBM MQ 创建的许多 System V IPC 资源可供机器上的所有用户访问，并且具有 -rw-rw-rw-许可权。

crtmqm 命令上的 **-g** *ApplicationGroup* 参数可用于将对队列管理器的访问限制为特定操作系统组的成员资格。 使用此受限组功能将进一步限制对 System V IPC 资源授予的许可权。

Linux AIX /opt/mqm 中具有 mqm 的 setuid 的 IBM MQ 文件许可权

以下信息涵盖了安全团队在目录树 $MQ_INSTALLATION_PATH 中标记某些可执行 IBM MQ 文件的情况，这违反了本地安全策略。 AIX 中的缺省位置为 /usr/mqm，其他 UNIX 操作系统的缺省位置为 /opt/mqm。 如果已在非缺省目录 (例如 /opt/mqm90) 中安装 IBM MQ，或者具有多个安装，那么本主题中的详细信息仍适用。

问题的原因

您的安全团队在 $*MQ_INSTALLATION_PATH* 下确定了以下关注领域:

1. /opt/mqm/bin 目录中的文件是其所在目录树的所有者的 setuid。 例如:

```
dr-xr-xr-x   mqm  mqm   ${MQ_INSTALLATION_PATH}/bin
-r-sr-s---   mqm  mqm   ${MQ_INSTALLATION_PATH}/bin/addmqinf
-r-sr-s---   mqm  mqm   ${MQ_INSTALLATION_PATH}/bin/amqcrsta
-r-sr-s---   mqm  mqm   ${MQ_INSTALLATION_PATH}/bin/amqfcxba
...
```

2. 几乎所有目录和文件都由 "mqm:mqm" 拥有，但以下目录和文件由 root 用户拥有:

```
dr-xr-x---   root mqm   ${MQ_INSTALLATION_PATH}/bin/security
-r-sr-x---   root mqm   ${MQ_INSTALLATION_PATH}/bin/security/amqoamax
-r-sr-x---   root mqm   ${MQ_INSTALLATION_PATH}/bin/security/amqoampx
```

   此子目录需要由 root 用户拥有，因为这些是来自 IBM MQ 客户机的用户指定密码时与操作系统交互的可执行文件，并且此密码由 IBM MQ 队列管理器传递到操作系统以确认密码是否有效。

3. 用户不拥有 /opt/mqm/lib/iconv directory 中的文件 (此目录在 AIX 上不存在)。 例如:

```
dr-xr-xr-x   mqm  mqm   ${MQ_INSTALLATION_PATH}/lib/iconv
-r--r--r--   bin  bin   ${MQ_INSTALLATION_PATH}/lib/iconv/002501B5.tbl
-r--r--r--   bin  bin   ${MQ_INSTALLATION_PATH}/lib/iconv/002501F4.tbl
-r--r--r--   bin  bin   ${MQ_INSTALLATION_PATH}/lib/iconv/00250333.tbl
...
```

4. 基于 RPM 的 Linux 系统上的修订包维护目录。 安装修订包时，现有文件将以类似于以下示例中所示的结构保存在此目录下，但在此示例中， V.R 表示 IBM MQ 版本和发行版号以及显示为依赖于已安装的修订包的子目录:

```
drwx------   root root  ${MQ_INSTALLATION_PATH}/maintenance
drwxr-xr-x   root root  ${MQ_INSTALLATION_PATH}/maintenance/V.R.0.1
drwxr-xr-x   root root  ${MQ_INSTALLATION_PATH}/maintenance/V.R.0.3
drwxr-xr-x   root root  ${MQ_INSTALLATION_PATH}/maintenance/V.R.0.4
...
```

解决问题

UNIX 系统上与 setuid 程序相关的一个问题是，可能会通过处理环境变量 (例如 LD* (LD_LIBRARY_PATH , AIX 上的 LIBPATH 等)) 来损害系统安全性。 这不再是一个问题，因为在装入 setuid 程序时， 各种 UNIX 操作系统现在会忽略这些 LD* 环境变量。

1. 为什么某些 IBM MQ 程序是 mqm-setuid 或 mqm-setgid。

   在 IBM MQ 中，用户标识 "mqm" 和属于 "mqm" 组的任何标识都是 IBM MQ 管理用户。

   通过对此用户进行认证来保护 IBM MQ 队列管理器资源。 由于队列管理器进程使用和修改这些队列管理器资源，因此队列管理器进程需要 "mqm" 权限才能访问这些资源。 因此， IBM MQ 队列管理器支持进程设计为使用有效用户标识 "mqm" 运行。

   为了帮助非管理用户访问 IBM MQ 对象， IBM MQ 提供了对象权限管理器 (OAM) 设施，在此设施中，可以根据非管理用户运行的应用程序的需要授予和撤销权限。

   由于能够为用户授予不同级别的认证，并且 **setuid** 和 **setgid** 程序忽略 LD* 变量，因此 IBM MQ 二进制文件和库文件不会以任何方式损害系统的安全性。

2. 无法在不影响 IBM MQ 功能的情况下更改许可权以满足企业的安全策略。

   不得更改任何 IBM MQ 二进制文件和库的许可权和所有权。 由于此类更改， IBM MQ 功能可能会受到影响，因此队列管理器进程可能无法访问某些资源。

   请注意，许可权和所有权不会对系统造成任何安全威胁。

   不得使用 nosuid 选项安装安装 IBM MQ 或 IBM MQ 数据所在的 Linux 硬盘驱动器/磁盘。 此配置可能会禁止 IBM MQ 功能。

   有关更多信息，请参阅第 59 页的『IBM MQ 文件系统许可权已应用于 /var/mqm』 。

相关概念

文件系统

Windows Windows 上的 IBM MQ 文件系统许可权

以下信息描述了应用于 Windows 上的文件和目录的安全性。 为了确保 IBM MQ 的正确操作，您不应更改 IBM MQ 设置的文件系统许可权。

数据目录

注: 在此目录的根目录上设置的许可权在整个目录结构中向下继承。

除以下文本中详述的异常外，数据目录 (DATADIR) 下的目录设置为具有以下许可权。

管理员
完全控制

mqm 组
完全控制

SYSTEM
完全控制

每个人
读和执行

例外情况如下所示:

DATADIR \errors
每个人都完全控制

DATADIR \trace
每个人都完全控制

DATADIR \log

管理员
完全控制

mqm 组
完全控制

SYSTEM
完全控制

每个人
读

DATADIR\log\<qmgrname>\active

管理员
完全控制

mqm 组
完全控制

SYSTEM
完全控制

未向每个人授予访问权。

错误日志文件 AMQERR01.LOG 等等，不从其目录继承其安全设置，而是设置为 **每个人： 完全控制**。

产品的先前发行版

在 IBM MQ 8.0 之前的产品发行版中，缺省程序和缺省数据目录位于同一位置。

在最初安装在 IBM MQ 8.0 之前的任何安装中。 并且已安装到缺省位置，然后从该位置升级，数据目录和程序目录仍然位于同一位置 (在 `C:\Program Files\IBM\WebSphere MQ` 中)。

对于位于同一位置的数据和程序目录，上述信息仅适用于属于数据目录的目录，而不适用于属于程序目录的目录。

队列的命名限制

队列名称的长度存在限制。 某些队列名称是为队列管理器定义的队列保留的。

名称长度限制

队列的名称长度最多可以为 48 个字符。

保留队列名称

将为队列管理器定义的队列保留以 “SYSTEM.” 开头的名称。 您可以使用 **ALTER** 或 **DEFINE REPLACE** 命令来更改这些队列定义以适合您的安装。 为 IBM MQ 定义了以下名称:

表 *41:* 保留队列名称和描述

队列名称	描述
SYSTEM.ADMIN.ACTIVITY.QUEUE	活动报告的队列
SYSTEM.ADMIN.CHANNEL.EVENT	通道事件的队列
SYSTEM.ADMIN.COMMAND.EVENT	命令事件的队列
SYSTEM.ADMIN.COMMAND.QUEUE	将 PCF 命令消息发送到的队列
SYSTEM.ADMIN.CONFIG.EVENT	配置事件的队列
SYSTEM.ADMIN.PERFM.EVENT	性能事件的队列
SYSTEM.ADMIN.PUBSUB.EVENT	系统发布/预订相关事件队列
SYSTEM.ADMIN.QMGR.EVENT	队列管理器事件的队列
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	跟踪路由应答消息的队列
SYSTEM.AUTH.DATA.QUEUE	用于保存队列管理器的访问控制表的队列。 (不适用于 z/OS)
SYSTEM.CHANNEL.INITQ	通道的启动队列
SYSTEM.CHANNEL.SYNCQ	保存通道同步数据的队列
SYSTEM.CHLAUTH.DATA.QUEUE	IBM MQ 通道认证数据队列
SYSTEM.CICS.INITIATION.QUEUE	用于触发的队列 (不适用于 z/OS)
SYSTEM.CLUSTER.COMMAND.QUEUE	用于在队列管理器之间传递版本库变更的队列
SYSTEM.CLUSTER.HISTORY.QUEUE	此队列用于存储集群状态信息的历史记录以用于服务目的。
SYSTEM.CLUSTER.REPOSITORY.QUEUE	用于保存资源库信息的队列
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	用于为每个集群发送方通道创建单个传输队列的队列。
SYSTEM.CLUSTER.TRANSMIT.QUEUE	群组支持管理的所有目的地的传输队列
SYSTEM.COMMAND.INPUT	在 z/OS 上向其发送命令消息的队列
SYSTEM.COMMAND.REPLY.MODEL	命令应答的模型队列定义 (对于 z/OS)
SYSTEM.DEAD.LETTER.QUEUE	死信队列 (不适用于 z/OS)
SYSTEM.DEFAULT.ALIAS.QUEUE	缺省别名队列定义
SYSTEM.DEFAULT.INITIATION.QUEUE	用于触发指定进程的队列 (不适用于 z/OS)
SYSTEM.DEFAULT.LOCAL.QUEUE	缺省本地队列定义
SYSTEM.DEFAULT.MODEL.QUEUE	缺省模型队列定义
SYSTEM.DEFAULT.REMOTE.QUEUE	缺省远程队列定义

表 41: 保留队列名称和描述 (继续)	
队列名称	**描述**
SYSTEM.DURABLE.SUBSCRIBER.QUEUE	用于在队列管理器中保存持久预订的持久副本的本地队列
SYSTEM.HIERARCHY.STATE	用于保存有关发布/预订层次结构中的队列间管理器关系状态的信息的队列
SYSTEM.JMS.TEMPQ.MODEL	JMS 临时队列的模型
SYSTEM.INTERNAL.REPLY.QUEUE	IBM MQ 内部应答队列 (不适用于 z/OS)
SYSTEM.INTER.QMGR.CONTROL	在发布/预订层次结构中用于接收来自远程队列管理器的请求以创建代理预订的队列
SYSTEM.INTER.QMGR.PUBS	在发布/预订层次结构中用于从远程队列管理器接收发布的队列
SYSTEM.INTER.QMGR.FANREQ	在发布/预订层次结构中用于处理请求以在远程队列管理器上创建代理预订的队列
SYSTEM.MQEXPLORER.REPLY.MODEL	IBM MQ Explorer 的应答的模型队列定义
SYSTEM.MQSC.REPLY.QUEUE	MQSC 命令应答的模型队列定义 (不适用于 z/OS)
SYSTEM.QSG.CHANNEL.SYNCQ	用于存储包含共享通道同步信息的消息的共享本地队列 (仅限 z/OS)
SYSTEM.QSG.TRANSMIT.QUEUE	在同一队列共享组中的队列管理器之间传输消息时，组内排队代理使用的共享本地队列 (仅限 z/OS)
SYSTEM.RETAINED.PUB.QUEUE	用于在队列管理器中保存每个保留发布的副本的本地队列。
SYSTEM.SELECTION.EVALUATION.QUEUE	IBM MQ 内部选择评估队列 (不适用于 z/OS)
SYSTEM.SELECTION.VALIDATION.QUEUE	IBM MQ 内部选择验证队列 (不适用于 z/OS)

其他对象的命名限制

对象名的长度存在限制。 某些对象名是为队列管理器定义的对象保留的。

名称长度限制

进程，名称列表，集群，主题，服务和认证信息对象的名称长度最多可以为 48 个字符。

通道的名称长度最多可以为 20 个字符。

存储类的名称长度最多可以为 8 个字符。

CF 结构的名称长度最多可以为 12 个字符。

保留对象名

以 SYSTEM 开头的名称。 为队列管理器定义的对象保留。 您可以使用 **ALTER** 或 **DEFINE REPLACE** 命令来更改这些对象定义以适合您的安装。 为 IBM MQ 定义了以下名称:

表 42: 保留对象名和描述	
对象名称	**描述**
SYSTEM.ADMIN.SVRCONN	用于队列管理器远程管理的服务器连接通道
SYSTEM.AUTO.RECEIVER	用于自动定义的缺省接收方通道 (仅限 AIX, Linux, and Windows 系统)

表 42: 保留对象名和描述 (继续)	
对象名称	**描述**
SYSTEM.AUTO.SVRCONN	用于自动定义的缺省服务器连接通道 (仅限多平台)
SYSTEM.BASE.TOPIC	ASPARENT 解析的基本主题。 如果特定管理主题对象没有父管理主题对象，那么将从此对象继承任何 ASPARENT 属性
SYSTEM.DEF.CLNTCONN	缺省客户机连接通道定义
SYSTEM.DEF.CLUSRCVR	缺省集群接收方通道定义
SYSTEM.DEF.CLUSSDR	缺省集群发送方通道定义
SYSTEM.DEF.RECEIVER	缺省接收方通道定义
SYSTEM.DEF.REQUESTER	缺省请求者通道定义
SYSTEM.DEF.SENDER	缺省发送方通道定义
SYSTEM.DEF.SERVER	缺省服务器通道定义
SYSTEM.DEF.SVRCONN	缺省服务器连接通道定义
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	用于定义类型为 CRLLDAP 的认证信息对象的缺省认证信息对象定义
SYSTEM.DEFAULT.AUTHINFO.OCSP	用于定义 OCSP 类型的认证信息对象的缺省认证信息对象定义
SYSTEM.DEFAULT.LISTENER.LU62	缺省 SNA 侦听器 (仅限 Windows)
SYSTEM.DEFAULT.LISTENER.NETBIOS	缺省 NetBIOS 侦听器 (仅限 Windows)
SYSTEM.DEFAULT.LISTENER.SPX	缺省 SPX 侦听器 (仅限 Windows)
SYSTEM.DEFAULT.LISTENER.TCP	缺省 TCP/IP 侦听器 (仅限多平台)
SYSTEM.DEFAULT.NAMELIST	缺省名称列表定义
SYSTEM.DEFAULT.PROCESS	缺省进程定义
SYSTEM.DEFAULT.SEVICE	缺省服务 (仅限多平台)
SYSTEM.DEFAULT.TOPIC	缺省主题定义
SYSTEM.QPUBSUB.QUEUE.NAMELIST	要监视的已排队发布/预订接口的队列列表
SYSTEMST	缺省存储类定义 (仅限 z/OS)

队列名称解析

在更大的网络中，使用队列管理器比其他形式的通信具有许多优点。 这些优点源自分布式队列管理中的名称解析功能，该功能可确保队列管理器在通道的发送端和接收端都执行队列名称解析。

这一办法的主要好处如下:

- 应用程序不需要进行路由决策
- 应用程序不需要知道网络结构
- 网络链接由系统管理员创建
- 网络结构由网络规划者控制
- 可以在节点之间使用多个通道来分区流量

下图显示了队列名称解析的示例。 该图显示网络中的两台机器，一台运行 put 应用程序，另一台运行 get 应用程序。 应用程序通过受 MCA 控制的 IBM MQ 通道相互通信。

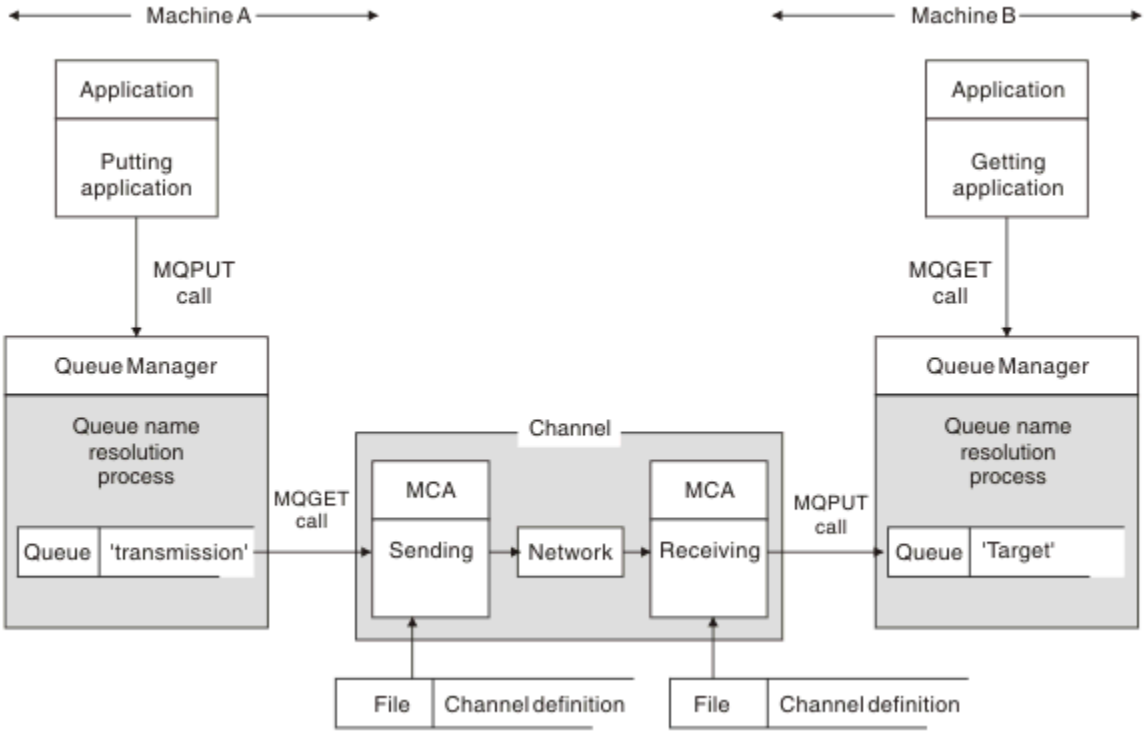

图 *5:* 名称解析

对于 第 68 页的图 5，对于应用程序而言，用于将消息放入远程队列的基本机制与用于将消息放入本地队列的基本机制相同:

- 放入消息的应用程序发出 MQOPEN 和 MQPUT 调用以将消息放入目标队列。
- 获取消息的应用程序发出 MQOPEN 和 MQGET 调用以从目标队列获取消息。

如果两个应用程序都连接到同一个队列管理器，那么不需要队列管理器间通信，并且目标队列对这两个应用程序都描述为 本地 。

但是，如果应用程序连接到不同的队列管理器，那么在传输中涉及到两个 MCA 及其关联的网络连接，如图所示。 在这种情况下，目标队列被视为放置应用程序的 远程队列 。

事件顺序如下所示:

1. 放置应用程序发出 MQOPEN 和 MQPUT 调用以将消息放置到目标队列。
2. 在 MQOPEN 调用期间， *name resolution* 函数检测到目标队列不是本地队列，并决定适合哪个传输队列。 此后，在与 MQOPEN 调用关联的 MQPUT 调用上，所有消息都放置在此传输队列上。
3. 发送 MCA 从传输队列中获取消息，并将它们传递到远程计算机上的接收 MCA。
4. 接收 MCA 将消息放在一个或多个目标队列上。
5. 获取应用程序发出 MQOPEN 和 MQGET 调用以从目标队列获取消息。

注: 只有步骤 1 和步骤 5 涉及应用程序代码; 步骤 2 到 4 由本地队列管理器和 MCA 程序执行。 放置应用程序不知道目标队列的位置，该队列可能位于同一处理器中，也可能位于另一个大陆上的另一个处理器中。

发送 MCA ，网络连接和接收 MCA 的组合称为 消息通道，本质上是单向设备。 正常情况下，需要双向移动消息，为此移动设置了两个通道，每个方向一个通道。

相关任务

将消息放在远程队列上

什么是队列名称解析?

队列名称解析对于分布式队列管理至关重要。 它使应用程序不再需要关注队列的物理位置，并使应用程序与网络的详细信息相隔离。

系统管理员可以将队列从一个队列管理器移至另一个队列管理器，并在无需应用程序了解任何信息的情况下更改队列管理器之间的路由。

要从应用程序设计中取消耦合数据传递的确切路径，在应用程序引用目标队列时使用的名称与发生流的通道的命名之间存在某种程度的不相关。 此双向是使用队列名称解析机制实现的。

实质上，当应用程序引用队列名称时，解析机制会将该名称映射到传输队列或非传输队列的本地队列。 对于到传输队列的映射，需要在目标处进行第二个名称解析，并且将接收到的消息按应用程序设计者的期望放在目标队列上。 应用程序仍然不知道用于移动消息的传输队列和通道。

注: 队列和通道的定义是系统管理责任，可以由操作员或系统管理实用程序进行更改，而无需更改应用程序。

消息流的系统管理的一个重要要求是需要在队列管理器之间提供备用路径。 例如，业务需求可能指示通过不同通道将不同的 服务类 发送到同一目标。 此决策是系统管理决策，队列名称解析机制提供了实现此决策的灵活方法。 The Application Programming Guide describes this in detail, but the basic idea is to use queue name resolution at the sending queue manager to map the queue name supplied by the application to the appropriate transmission queue for the type of traffic involved. 同样在接收端，队列名称解析将消息描述符中的名称映射到本地 (非传输) 队列或再次映射到相应的传输队列。

不仅可以将从一个队列管理器到另一个队列管理器的正向路径分区为不同类型的流量，而且发送到出站消息中的应答队列定义的返回消息也可以使用相同的流量分区。 队列名称解析满足此要求，并且应用程序设计者无需参与这些流量分区决策。

在发送和接收队列管理器上执行映射的点是名称解析工作方式的一个重要方面。 此映射允许将放置应用程序提供的队列名称映射到发送队列管理器上的本地队列或传输队列，并再次重新映射到接收队列管理器上的本地队列或传输队列。

来自接收应用程序或 MCA 的应答消息具有以相同方式执行的名称解析，允许使用路由上所有队列管理器中的队列定义在特定路径上进行返回路由。

如何解析别名，远程队列和集群队列的目标对象属性?

代表应用程序 API 调用执行名称解析时，将从最初指定的对象， “path” (请参阅 第 67 页的『队列名称解析』) 和已解析的目标对象的组合中解析影响对象使用的属性。 在队列管理器集群中，有问题的 “named object” 是集群对象 (队列或主题) 定义。 这是队列管理器之间共享且可视的对象属性的子集。 例如，**DISPLAY QCLUSTER**。

如果可以在应用程序打开的指定对象上定义属性，那么此属性优先。 例如，可以在别名和远程队列定义上配置所有 DEF* *** 属性 (缺省持久性，优先级和异步放置响应)。 当应用程序打开别名或远程队列 (而不是任何已解析的目标队列或传输队列) 时，这些操作将生效。

通常无法在指定对象 (远程队列定义或别名) 上定义旨在限制或限制应用程序与目标对象的交互的属性。 例如，无法在远程队列定义或别名上设置 **MAXMSGL** 和 **MAXDEPTH** ，并且不会在队列管理器集群的成员之间传递。 因此，将从已解析的队列 (例如，本地队列，相应的传输队列或 SYSTEM.CLUSTER.TRANSMIT.QUEUE)。 到达远程队列管理器时，可能会在传递到目标队列时应用第二个约束，这可能导致消息被放置在死信队列上，或者通道被强制停止。

请注意，属性解析的特殊情况是 **PUT** 和 **GET** 启用。 对于这两个属性，队列路径中的任何 **DISABLED** 实例都会导致整体解析属性 **DISABLED**。

系统和缺省对象

列出由 **crtmqm** 命令创建的系统和缺省对象。

使用 **crtmqm** 控制命令创建队列管理器时，将自动创建系统对象和缺省对象。

- 系统对象是操作队列管理器或通道所需的那些 IBM MQ 对象。

- 缺省对象定义对象的所有属性。 创建对象 (例如本地队列) 时，未显式指定的任何属性都将从缺省对象继承。

下表列出了 **crtmqm** 创建的系统对象和缺省对象。

注: 还有两个缺省对象未包含在表中: 队列管理器对象和对象目录。 这些是记录和可恢复的对象。

- 系统和缺省对象: 队列
- 系统对象和缺省对象: 主题
- 系统和缺省对象: 服务器通道
- 系统和缺省对象: 客户机通道
- 系统对象和缺省对象: 认证信息
- 系统和缺省对象: 通信信息
- 系统和缺省对象: 侦听器
- 系统对象和缺省对象: 名称列表
- 系统和缺省对象: 进程
- 系统和缺省对象: 服务

表 *43:* 系统和缺省对象*:* 队列

对象名	**描述**
SYSTEM.ADMIN.ACCOUNTING.QUEUE	当应用程序与队列管理器断开连接时生成的记帐消息数据的队列。
SYSTEM.ADMIN.ACTIVITY.QUEUE	用于保存返回的活动报告消息的队列。
SYSTEM.ADMIN.CHANNEL.EVENT	通道的事件队列。
SYSTEM.ADMIN.COMMAND.EVENT	命令事件的事件队列。
SYSTEM.ADMIN.COMMAND.QUEUE	管理命令队列。 用于远程 MQSC 命令和 PCF 命令。
SYSTEM.ADMIN.CONFIG.EVENT	配置事件的事件队列。
SYSTEM.ADMIN.LOGGER.EVENT	记录器事件 (日志接收器) 消息的事件队列。
SYSTEM.ADMIN.PERFM.EVENT	性能事件的事件队列。
SYSTEM.ADMIN.PUBSUB.EVENT	系统发布/预订相关事件队列
SYSTEM.ADMIN.QMGR.EVENT	队列管理器事件的事件队列。
SYSTEM.ADMIN.STATISTICS.QUEUE	用于保存 MQI，队列和通道统计信息监视数据的队列。
SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE	显示跟踪活动的队列。
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	用于保存返回的跟踪路由应答消息的队列。
SYSTEM.AMQP.COMMAND.QUEUE	IBM MQ AMQP 管理命令队列
SYSTEM.AUTH.DATA.QUEUE	用于保存队列管理器的访问控制表的队列。 由对象权限管理器 (OAM) 使用。
SYSTEM.BROKER.ADMIN.STREAM	已排队的发布/预订接口的管理流
SYSTEM.BROKER.CONTROL.QUEUE	发布/预订接口控制队列。
SYSTEM.BROKER.DEFAULT.STREAM	已排队的发布/预订接口的缺省流
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	代理到代理通信队列。
SYSTEM.CHANNEL.INITQ	通道启动队列。

表 *43:* 系统和缺省对象*:* 队列 *(*继续*)*	
对象名	**描述**
SYSTEM.CHANNEL.SYNCQ	用于保存通道的同步数据的队列。
SYSTEM.CHLAUTH.DATA.QUEUE	IBM MQ 通道认证数据队列
SYSTEM.CICS.INITIATION.QUEUE	缺省 CICS 启动队列。
SYSTEM.CLUSTER.COMMAND.QUEUE	用于将消息传递到存储库队列管理器的队列。
SYSTEM.CLUSTER.HISTORY.QUEUE	用于存储集群状态信息的历史记录以用于服务目的的队列。
SYSTEM.CLUSTER.REPOSITORY.QUEUE	用于存储所有存储库信息的队列。
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	用于为每个集群发送方通道创建单个传输队列的队列。
SYSTEM.CLUSTER.TRANSMIT.QUEUE	所有消息到所有集群的传输队列。
SYSTEM.DEAD.LETTER.QUEUE	死信 (undelivered-message) 队列。
SYSTEM.DEFAULT.ALIAS.QUEUE	缺省别名队列。
SYSTEM.DEFAULT.INITIATION.QUEUE	缺省启动队列。
SYSTEM.DEFAULT.LOCAL.QUEUE	缺省本地队列。
SYSTEM.DEFAULT.MODEL.QUEUE	缺省模型队列。
SYSTEM.DEFAULT.REMOTE.QUEUE	缺省远程队列。
SYSTEM.DOTNET.XARECOVERY.QUEUE	IBM MQ .NET XA 恢复队列
SYSTEM.DURABLE.MODEL.QUEUE	用作受管持久预订模型的队列。
SYSTEM.DURABLE.SUBSCRIBER.QUEUE	用于在队列管理器中保存持久预订的持久副本的队列。
SYSTEM.HIERARCHY.STATE	IBM MQ 分布式发布/预订层次结构关系状态。
SYSTEM.INTER.QMGR.CONTROL	IBM MQ 分布式发布/预订控制队列。
SYSTEM.INTER.QMGR.FANREQ	IBM MQ 分布式发布/预订内部代理预订扇出进程输入队列。
SYSTEM.INTER.QMGR.PUBS	IBM MQ 分布式发布/预订发布。
SYSTEM.INTERNAL.REPLY.QUEUE	
SYSTEM.INTERNAL.REQUEST.QUEUE	
SYSTEM.JMS.TEMPQ.MODEL	JMS 临时队列的模型
SYSTEM.MQEXPLORER.REPLY.MODEL	IBM MQ Explorer 应答队列。 这是一个模型队列，用于为对 IBM MQ Explorer 的应答创建临时动态队列。
SYSTEM.MQSC.REPLY.QUEUE	MQSC 命令应答队列。 这是一个模型队列，用于为远程 MQSC 命令的应答创建临时动态队列。
SYSTEM.NDURABLE.MODEL.QUEUE	用作受管非持久预订模型的队列。
SYSTEM.PENDING.DATA.QUEUE	支持 JMS 中的延迟消息。
SYSTEM.PROTECTION.ERROR.QUEUE	IBM MQ 消息保护错误队列。
SYSTEM.PROTECTION.POLICY.QUEUE	IBM MQ 消息保护策略队列。
SYSTEM.REST.REPLY.QUEUE	
SYSTEM.RETAINED.PUB.QUEUE	用于在队列管理器中保存每个保留发布的副本的队列。

表 *43:* 系统和缺省对象*:* 队列 *(*继续*)*	
对象名	**描述**
SYSTEM.SELECTION.EVALUATION.QUEUE	
SYSTEM.SELECTION.VALIDATION.QUEUE	

表 *44:* 系统和缺省对象*:* 主题	
对象名	**描述**
SYSTEM.ADMIN.TOPIC	管理主题。
SYSTEM.BASE.TOPIC	ASPARENT 解析的基本主题。 如果特定主题没有父管理主题对象，或者这些父对象也具有 ASPARENT，那么将从此对象继承任何剩余的 ASPARENT 属性。
SYSTEM.BROKER.ADMIN.STREAM	由排队的发布/预订接口使用的管理流。
SYSTEM.BROKER.DEFAULT.STREAM	已排队的发布/预订接口所使用的缺省流。
SYSTEM.BROKER.DEFAULT.SUBPOINT	已排队的发布/预订接口使用的缺省子点。
SYSTEM.DEFAULT.TOPIC	缺省主题定义。

表 *45:* 系统和缺省对象*:* 服务器通道	
对象名	**描述**
SYSTEM.AUTO.RECEIVER	动态接收方通道。
SYSTEM.AUTO.SVRCONN	动态服务器连接通道。
SYSTEM.DEF.AMQP	缺省 AMQP 通道。 请注意，已定义对象，但不支持 AMQP 服务。
SYSTEM.DEF.CLUSRCVR	集群的缺省接收方通道，用于为在集群中的队列管理器上创建 CLUSRCVR 通道时未指定的任何属性提供缺省值。
SYSTEM.DEF.CLUSSDR	集群的缺省发送方通道，用于提供在集群中的队列管理器上创建 CLUSSDR 通道时未指定的任何属性的缺省值。
SYSTEM.DEF.RECEIVER	缺省接收方通道。
SYSTEM.DEF.REQUESTER	缺省请求者通道。
SYSTEM.DEF.SENDER	缺省发送方通道。
SYSTEM.DEF.SERVER	缺省服务器通道。
SYSTEM.DEF.SVRCONN	缺省服务器连接通道。
SYSTEM.DEFAULT.AUTHINFO.IDPWLDAP	
SYSTEM.DEFAULT.AUTHINFO.IDPWOS	

表 *46:* 系统和缺省对象*:* 客户机通道	
对象名	**描述**
SYSTEM.DEF.CLNTCONN	缺省客户机连接通道。

表 *47:* 系统和缺省对象*:* 认证信息

对象名	**描述**
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	用于定义类型为 CRLLDAP 的认证信息对象的缺省认证信息对象。
SYSTEM.DEFAULT.AUTHINFO.OCSP	用于定义 OCSP 类型的认证信息对象的缺省认证信息对象。

表 *48:* 系统和缺省对象*:* 通信信息

对象名	**描述**
SYSTEM.DEFAULT.COMMINFO.MULTICAST	多点广播的缺省通信信息对象。

表 *49:* 系统和缺省对象*:* 侦听器

对象名	**描述**
SYSTEM.DEFAULT.LISTENER.TCP	TCP 传输的缺省侦听器。
Windows SYSTEM.DEFAULT.LISTENER.LU62	缺省 LU62 侦听器。
Windows SYSTEM.DEFAULT.LISTENER.NETBIOS	缺省 NETBIOS 侦听器。
Windows SYSTEM.DEFAULT.LISTENER.SPX	缺省 SPX 侦听器。

表 *50:* 系统对象和缺省对象*:* 名称列表

对象名	**描述**
SYSTEM.DEFAULT.NAMELIST	缺省名称列表定义。
SYSTEM.QPUBSUB.QUEUE.NAMELIST	由排队的发布/预订接口监视的队列名称列表。
SYSTEM.QPUBSUB.SUBPOINT.NAMELIST	由排队的发布/预订接口用于将主题对象与预订点匹配的主题对象的列表。

表 *51:* 系统和缺省对象*:* 进程

对象名	**描述**
SYSTEM.DEFAULT.PROCESS	缺省进程定义。

表 *52:* 系统和缺省对象*:* 服务

对象名	**描述**
SYSTEM.AMQP.SERVICE	MQ Light API 服务。 请注意，已定义对象，但不支持该服务。
SYSTEM.DEFAULT.SERVICE	缺省服务。

SYSTEM.BASE.TOPIC

ASPARENT 解析的基本主题。 如果特定主题没有父管理主题对象，或者这些父对象也具有 ASPARENT，那么将从此对象继承任何剩余的 ASPARENT 属性。

表 53: *SYSTEM.BASE.TOPIC* 的缺省值	
参数	**值**
TOPICSTR	"
V 9.4.0 CAPEXPRY	NOLIMIT
CLROUTE	DIRECT
cluster	缺省值为空字符串。
COMMINFO	SYSTEM.DEFAULT.COMMINFO.MULTICAST
DEFPRESP	SYNC
DEFPRTY	0
DEFPSIST	NO
DESCR	"用于解析属性的基本主题"
DURSUB	YES
MCAST	DISABLED
MDURMDL	SYSTEM.DURABLE.MODEL.QUEUE
MNDURMDL	SYSTEM.NDURABLE.MODEL.QUEUE
NPMSGDLV	ALLAVAIL
PMSGDLV	ALLDUR
PROXYSUB	FIRSTUSE
PUB	ENABLED
PUBSCOPE	ALL
z/OS z/OS QSGDISP (仅限 z/OS 平台)	QMGR
SUB	ENABLED
SUBSCOPE	ALL
USEDLQ	YES
WILDCARD	PASSTHRU

如果此对象不存在，那么其缺省值仍由 IBM MQ 用于 ASPARENT 属性，而这些属性未由主题树上的父主题解析。

将 SYSTEM.BASE.TOPIC 的 PUB 或 SUB 属性设置为 DISABLED 会阻止应用程序发布或预订主题树中的主题，但有两个例外:

1. 主题树中 PUB 或 SUB 显式设置为 ENABLE 的任何主题对象。 应用程序可以发布或预订这些主题及其子代。
2. 通过将 SYSTEM.BASE.TOPIC 的 PUB 或 SUB 属性设置为 DISABLED，不会禁用对 SYSTEM.BROKER.ADMIN.STREAM 的发布和预订。

另请参阅 **PUB** 参数的特殊处理。

配置文件节信息

以下信息可帮助您在节中配置信息，并列出 mqs.ini， qm.ini 和 mqclient.ini 文件的内容。

配置节

警告: Windows Windows 注册表包含关键信息，包括文件位置、驱动程序和启动进程。 IBM MQ 需要依赖这些特定的注册表设置才能正常运行。 在没有适当信息的情况下更改这些设置可能会导致 IBM MQ 应用程序无法使用或完全失效，因为不正确的更改会破坏其基本功能。

使用链接可帮助您在企业中配置一个或多个系统:

- mqs.ini 文件节和属性 可帮助您配置:
  - *AllQueueManagers* 一节诗
  - *DefaultQueueManager* 一节诗
  - *ExitProperties* 节
  - *LogDefaults* 节
  - `qm.ini` 文件中的 *Security* 节
- qm.ini 文件节和属性 可帮助您配置:
  - Windows *AccessMode* 节 (仅限 Windows)
  - *Service* 节-用于可安装服务
  - *Log* 节
  - Linux AIX *RestrictedMode* 节 (仅限 AIX and Linux 系统)
  - *XAResourceManager* 节
  - *TCP*， *LU62* 和 *NETBIOS* 节
  - *ExitPath* 节
  - *QMErrorLog* 节
  - *SSL* 节
  - *ExitPropertiesLocal* 一节诗
- 配置服务和组件 可帮助您配置:
  - *Service* 节
  - *ServiceComponent* 节

  并包含用于 AIX, Linux, and Windows 平台上不同服务的方式的链接。
- 配置 API 出口 可帮助您配置以下内容:
  - *AllActivityTrace* 一节诗
  - *AppplicationTrace* 节
- 配置活动跟踪行为 可帮助您配置:
  - *ApiExitCommon* 一节诗
  - *ApiExitTemplate* 一节诗
  - *ApiExitLocal* 一节诗
- IBM MQ MQI client 配置文件 `mqclient.ini` 可帮助您配置:
  - *CHANNELS* 节
  - *ClientExitPath* 一节诗
  - Windows *LU62*， *NETBIOS* 和 *SPX* 节 (仅限 Windows)
  - *MessageBuffer* 节
  - *SSL* 节
  - *TCP* 节
  - V 9.4.0 *trace* 节 (仅用于 IBM MQ .NET 和 XMS .NET)

- 第 77 页的『用于分布式排队的配置文件节』 可帮助您配置:
  - *CHANNELS* 节
  - *TCP* 节
  - *LU62* 节
  - *NETBIOS*
  - *ExitPath* 节
- 设置已排队的发布/预订消息属性 可帮助您配置:
  - *PersistentPublishRetry* 特质
  - *NonPersistentPublishRetry* 特质
  - *PublishBatchSize* 特质
  - *PublishRetryInterval* 特质

  在 *Broker* 节中。

  注意: 如果需要 *Broker* 节，那么必须创建一个节。
- 使用自动配置可帮助您配置:
  - AutoConfig 节
  - AutoCluster 节
  - Variables 节

配置文件

请参阅:

- **mqs.ini** 文件
- **qm.ini** 文件
- **mqclient.ini** 文件

以获取每个配置文件中可能的节的列表。 Linux AIX

mqs.ini 文件

AIX and Linux 系统的 IBM MQ 配置文件示例 显示了示例 `mqs.ini` 文件。

`mqs.ini` 文件可以包含以下节:

- *AllQueueManagers*
- *DefaultQueueManager*
- *ExitProperties*
- *LogDefaults*

此外，每个队列管理器都有一个 *QueueManager* 节。

qm.ini 文件

IBM MQ for AIX or Linux 系统的示例队列管理器配置文件 显示了示例 `qm.ini` 文件。

`qm.ini` 文件可以包含以下节:

- *ExitPath*
- 日志
- *QMErrorLog*
- *QueueManager*
- 安全性

- *ServiceComponent*

  Multi 要配置 *InstallableServices*，请使用 服务 和 *ServiceComponent* 节。
- 连接 *DefaultBindType*

  注意: 如果需要 *Connection* 节，那么必须创建一个节。
- *SSL* 和 *TLS*
- *TCP*， *LU62* 和 *NETBIOS*
- XAResourceManager

此外，您还可以更改以下内容:

- Windows *AccessMode* (仅限 Windows)
- Linux AIX *RestrictedMode* (仅限 AIX and Linux 系统)

使用 crtmqm 命令。

mqclient.ini 文件

`mqclient.ini` 文件可以包含以下节:

- 通道
- *ClientExitPath*
- *LU62*， *NETBIOS* 和 *SPX*
- *MessageBuffer*
- *SSL*
- *TCP*

此外，您可能需要 PreConnect 节 来配置预连接出口。

用于分布式排队的配置文件节

与分布式排队相关的队列管理器配置文件 qm.ini 节的描述。

本主题显示队列管理器配置文件中与分布式排队相关的节。 它适用于 IBM MQ for Multiplatforms 的队列管理器配置文件。 该文件在所有平台上都称为 `qm.ini` 。

与分布式排队相关的节为:

- 通道
- TCP
- LU62
- NETBIOS
- 出口路径

第 78 页的图 6 显示了可以使用这些节设置的值。 在定义其中一个节时，不需要在新行上启动每个项。 可以使用分号 (;) 或散列字符 (#) 来指示注释。

```
CHANNELS:
 MAXCHANNELS=n           ; Maximum number of channels allowed, the
                         ; default value is 100.
 MAXACTIVECHANNELS=n     ; Maximum number of channels allowed to be active at
                         ; any time, the default is the value of MaxChannels.
 MAXINITIATORS=n         ; Maximum number of initiators allowed, the default
                         ; and maximum value is 3.
 MQIBINDTYPE=type        ; Whether the binding for applications is to be
                         ; "fastpath" or "standard".
                         ; The default is "standard".
 PIPELINELENGTH=n        ; The maximum number of concurrent threads a channel will use.
                         ; The default is 1. Any value greater than 1 is treated as 2.
 ADOPTNEWMCA=chltype     ; Stops previous process if channel fails to start.
                         ; The default is "NO".
 ADOPTNEWMCATIMEOUT=n    ; Specifies the amount of time that the new
                         ; process should wait for the old process to end.
                         ; The default is 60.
 ADOPTNEWMCACHECK=       ; Specifies the type checking required.
          typecheck      ; The default is "NAME","ADDRESS", and "QM".
 CHLAUTHEARLYADOPT=Y/N   ; The order in which connection authentication and channel
authentication rules are
                         ; processed. If not present in the qm.ini file the default is "N".
From MQ9.0.4 all
                         ; queue managers are created with a defa\ult of "Y"
 PASSWORDPROTECTION=     ; From MQ8.0,set protected passwords in the MQCSP structure, rather
than using TLS.
             options     ; The options are "compatible", "always", "optional" and "warn"
                         ; The default is "compatible".
 IGNORESEQNUMBERMISMATCH ; How the queue manager handles a sequence number mismatch during
channel startup.
                =YES/NO  ; The options are "YES" and "NO" (case insensitive) with the default
being "NO".
 CHLAUTHIGNOREUSERCASE   ; Enables a queue manager to make username matching within CHLAUTH
rules case-insensitive.
                   =Y/N  ; The options are "Y" and "N" with the default being "N".
 CHLAUTHISSUEWARN=Y      ; If you want message AMQ9787 to be generated when you set theWARN=YES
attribute
                         ; on the SET CHLAUTH command.
TCP:                     ; TCP entries
 PORT=n                  ; Port number, the default is 1414
 KEEPALIVE=Yes           ; Switch TCP/IP KeepAlive on
LU62:
 LIBRARY2=DLLName2       ; Used if code is in two libraries
EXITPATH:[1]     Location of user exits
 EXITPATHS=              ; String of directory paths.
```

图 *6:* 用于分布式排队的 *qm.ini* 节

注意:

1. EXITPATH 仅适用于以下平台:
   - AIX
   - Windows

相关任务

配置

z/OS 配置 z/OS

在 Multiplatforms 版上更改 IBM MQ 配置信息

通道属性

本部分描述了在通道定义中保留的通道属性。

您可以选择要针对每个通道的特定环境集最佳的通道属性。 但是，当通道正在运行时，实际值在启动协商期间可能已更改。 请参阅 准备通道。

许多属性都具有缺省值，您可以将这些值用于大多数通道。 但是，在缺省值并非最佳的情况下，请参阅本节以获取有关选择正确值的指导信息。

注: 在 IBM MQ for IBM i 中，可以将大多数属性指定为 *SYSDFTCHL，这意味着该值取自系统中的系统缺省通道。

下表按 MQSC 命令参数的顺序列出了 IBM MQ 通道属性的通道类型。

注: 对于集群通道（表中的 CLUSSDR 和 CLUSRCVR 列），如果某个属性可以同时在两个通道上设置，请同时进行设置并确保两处的设置相同。 如果两处的设置之间存在任何差异，那么可能会使用您在 CLUSRCVR 通道上指定的设置。 在集群通道中对此进行了解释。

表 *54:* 通道类型的通道属性

属性字段	MQSC 命令参数	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	AMQP
连接亲缘关系	AFFINITY					Yes				
更改日期	ALTDATE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
变更时间	ALTTIME	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Multi AMQP 保持活动状态	AMQPKA									Yes
批处理脉动信号间隔	BATCHHB	Yes	Yes					Yes	Yes	
批处理时间间隔	BATCHINT	Yes	Yes					Yes	Yes	
批次限制	BATCHLIM	Yes	Yes					Yes	Yes	
批次大小	BATCHSZ	Yes	Yes	Yes	Yes			Yes	Yes	
证书标签	CERTLABL	Yes	Yes	Yes	Yes	Yes	Yes	是第 82 页的『1』	Yes	Yes
通道名称	通道	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
通道类型	CHLTYPE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
客户机通道权重	CLNTWGHT					Yes				
集群名称列表	CLUSNL							Yes	Yes	
聚类	聚类:							Yes	Yes	
集群工作负载优先级	CLWLPRTY							Yes	Yes	
集群工作负载列组	CLWLRANK							Yes	Yes	
集群工作负载权重	CLWLWGHT							Yes	Yes	
头压缩	COMPHDR	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
数据压缩	COMPMSG	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
连接名称	CONNAME	Yes	Yes		Yes	Yes		Yes	Yes	
转换消息	CONVERT	Yes	Yes					Yes	Yes	
缺省重新连接	DEFRECON					Yes				

表 54: 通道类型的通道属性 (继续)

属性字段	MQSC 命令参数	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	AMQP
描述	DESCR	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
断开连接时间间隔	DISCINT	Yes	Yes				z/OS 是 Multi 否	Yes	Yes	
脉动信号间隔	HBINT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
保持活动时间间隔	KAINT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
本地地址	LOCLADDR	Yes	Yes		Yes	Yes		Yes	Yes	Yes
长重试次数	LONGRTY	Yes	Yes					Yes	Yes	
长重试时间间隔	LONGTMR	Yes	Yes					Yes	Yes	
最大实例数	MAXINST						Yes			Yes
每个客户机的最大实例数	MAXINSTC						Yes			
最大消息长度	MAXMSGL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
消息通道代理程序名称	MCANAME	Yes	Yes		Yes			Yes	Yes	
消息通道代理程序类型	MCATYPE	Yes	Yes		Yes			Yes	Yes	
消息通道代理程序用户	MCAUSER	Yes	Yes	Yes	Yes		Yes	Yes	Yes	Yes
LU 6.2 方式名	MODENAME	Yes	Yes		Yes	Yes		Yes	Yes	
Monitoring	MONCHL	Yes	Yes	Yes	Yes		Yes	Yes	Yes	
消息重试出口用户数据	MRDATA			Yes	Yes				Yes	
消息重试出口名称	MREXIT			Yes	Yes				Yes	
消息重试次数	MRRTY			Yes	Yes				Yes	
消息重试时间间隔	MRTMR			Yes	Yes				Yes	
消息出口用户数据	MSGDATA	Yes	Yes	Yes	Yes			Yes	Yes	
消息出口名	MSGEXIT	Yes	Yes	Yes	Yes			Yes	Yes	
网络连接优先级	NETPRTY								Yes	
非持久消息速度	NPMSPEED	Yes	Yes	Yes	Yes			Yes	Yes	
密码	密码	Yes	Yes		Yes	Yes		Yes		
端口号	PORT									Yes
PROPCTL MQGMO 的通道选项	PROPCTL	Yes	Yes					Yes	Yes	

表 54: 通道类型的通道属性 (继续)

属性字段	MQSC 命令参数	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	AMQP
放入权限	PUTAUT			Yes	Yes		z/OS 是 Multi 否		Yes	
队列管理器名称	QMNAME					Yes				
z/OS 处置	QSGDISP	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
接收出口用户数据	RCVDATA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
接收出口用户名	RCVEXIT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
安全出口用户数据	SCYDATA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
安全出口名	SCYEXIT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
发送出口用户数据	SENDDATA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
发送出口名	SENDEXIT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
序号回绕	SEQWRAP	Yes	Yes	Yes	Yes			Yes	Yes	
共享连接	SHARECNV					Yes	Yes			
短重试次数	SHORTRTY	Yes	Yes					Yes	Yes	
短重试时间间隔	SHORTTMR	Yes	Yes					Yes	Yes	
z/OS 安全策略保护	SPLPROT	Yes	Yes	Yes	Yes					
SSL 客户机认证	SSLCAUTH		Yes	Yes	Yes		Yes		Yes	Yes
SSL 密码规范	SSLCIPH	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SSL 同级	SSLPEER	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
通道统计信息	STATCHL	Yes	Yes	Yes	Yes			Yes	Yes	
LU 6.2 事务程序名	TPNAME	Yes	Yes		Yes	Yes		Yes	Yes	
主题根目录	TPROOT									Yes
传输类型	TRPTYPE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
使用客户机标识	USECLTID									Yes
使用死信队列	USEDLQ	Yes	Yes	Yes	Yes			Yes	Yes	
用户标识	USERID	Yes	Yes		Yes	Yes		Yes		
传输队列的名称	XMITQ	Yes	Yes							

表 *54:* 通道类型的通道属性 *(*继续*)*

属性字段	MQSC 命令参数	SDR	SVR	RCVR	RQSTR	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	AMQP

注意:

1. 没有任何管理接口允许为 CLUSSDR 通道查询或设置此属性。 您将收到 MQRCCF_WRONG_CHANNEL_TYPE 消息。 但是，此属性存在于 CLUSSDR 通道对象 (包括 MQCD 结构) 中，如果需要， CHAD 出口可以通过程序设置此属性。

对于某些平台， IBM MQ 可能未实现此部分中显示的所有属性。 在个别属性描述中提及异常和平台差异 (如果相关)。

每个属性的名称都显示在方括号中。

这些属性按组的字母顺序排列。

相关参考

MQSC 命令

ALTER CHANNEL

DEFINE CHANNEL

MQSC 关键字 (A-B) 的通道属性

MQSC 关键字的通道属性的字母顺序列表，以字母 *A* 或 *B* 开头。

AFFINITY (连接亲缘关系)

此属性指定使用同一队列管理器名称多次连接的客户机应用程序是否使用同一客户机通道。

当有多个适用的通道定义可用时，请使用此属性 (MQIACH_CONNECTION_AFFINITY)。

可能的值:

PREFERRED
进程中读取客户机通道定义表的第一个连接根据客户机通道权重创建一个适用的定义列表，权重为 0 的任何定义按字母顺序排在前面。 进程中的每个连接尝试使用该列表中的第一个定义进行连接。 如果连接不成功，那么将使用下一个定义。 客户机通道权重值不为 0 的不成功定义将移至列表末尾。 客户机通道权重为 0 的定义仍位于列表的前面，每次连接时将首先选择这些定义。

具有相同主机名的每个客户机进程总是创建相同的列表。

对于以 C， C++ 或 .NET 编程框架 (包括完全受管 .NET) 编写的客户机应用程序以及使用 IBM MQ classes for Java 和 IBM MQ classes for JMS 的应用程序，如果自创建列表以来已修改 CCDT，那么将更新该列表。

此值是缺省值，其值为 1。

NONE
进程中读取 CCID 的第一个连接创建适用的定义列表。 进程中的所有连接根据客户机通道权重选择适用的定义，首先按字母顺序选择权重为 0 的任何定义。

对于以 C， C++ 或 .NET 编程框架 (包括完全受管 .NET) 编写的客户机应用程序以及使用 IBM MQ classes for Java 和 IBM MQ classes for JMS 的应用程序，如果自创建列表以来已修改 CCDT，那么将更新该列表。

此属性仅对客户机连接通道类型有效。

ALTDATE (变更日期)

此属性是上次更改定义的日期，格式为 `yyyy-mm-dd`，并且对所有通道类型都有效。

ALTTIME (变更时间)

此属性是上次更改定义的时间，格式为 hh.mm.ss ，并且对所有通道类型有效。

AMQPKA (AMQP 保持活动)

Multi

使用 **AMQPKA** 属性来指定 AMQP 客户机连接的保持活动时间。 如果 AMQP 客户机未在保持活动时间间隔内发送任何帧，那么将关闭连接。

AMQPKA 属性确定从 IBM MQ 发送到 AMQP 客户机的空闲超时属性的值。 此属性是一个以毫秒为单位的时间段。

如果 **AMQPKA** 设置为值> 0 ，那么 IBM MQ 会将该值的一半作为空闲超时属性。 例如，值 10000 将导致队列管理器发送空闲超时值 5000。 客户机应确保至少每 10000 毫秒将数据发送到 IBM MQ 。 如果 IBM MQ 在该时间内未接收到数据，那么 IBM MQ 将假定客户机已失去其连接，并强制关闭具有 amqp:resource-limit-exceeded 错误条件的连接。

值 AUTO 或 0 表示 IBM MQ 不会将空闲超时属性流至 AMQP 客户机。

AMQP 客户机仍可以传递其自己的空闲超时值。 如果这样做，那么 IBM MQ 会至少频繁地流动数据 (或空 AMQP 帧) ，以通知客户机它可用。

BATCHHB (批处理脉动信号间隔)

该属性允许发送信道在提交一批报文之前验证接收信道是否仍处于活动状态。

因此，批处理脉动信号间隔允许回退批处理，而不是在接收通道未处于活动状态时变为不确定状态。 通过回退批处理，这些消息仍可用于处理，例如，可以将它们重定向到另一个通道。

如果发送通道在批处理脉动信号间隔内具有来自接收通道的通信，那么假定接收通道仍处于活动状态，否则将 "脉动信号" 发送到接收通道以进行检查。 发送通道会根据通道脉动信号间隔 (HBINT) 属性中指定的秒数，等待来自通道接收端的响应一段时间。

该值以毫秒为单位，必须在范围 0 到 999999 之间。 值为零表示不使用批处理心脏跳动。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 集群发送方
- 集群接收方

BATCHINT (批处理时间间隔)

此属性是一个周期 (以毫秒为单位) ，在此期间通道保持批处理打开，即使传输队列上没有消息也是如此。

可以指定任何毫秒数，从零到 999 999 999。 缺省值为零。

如果未指定批处理时间间隔，那么当满足下列其中一个条件时，批处理将关闭:

- 已发送 BATCHSZ 中指定的消息数。
- 已发送 BATCHLIM 中指定的字节数。
- 传输队列为空。

在负载较轻的通道上，如果传输队列频繁变为空，那么有效批处理大小可能比 BATCHSZ 小得多。

您可以使用 BATCHINT 属性通过减少短批处理数来提高通道的效率。 但是，请注意，您可以减慢响应时间，因为批处理持续时间较长，而消息未落实的时间较长。

如果指定 BATCHINT ，那么仅当满足下列其中一个条件时，批处理才会关闭:

- 已发送 BATCHSZ 中指定的消息数。
- 已发送 BATCHLIM 中指定的字节数。

- 传输队列上没有更多消息，并且在等待消息时已经过 BATCHINT 时间间隔 (自检索到批处理的第一条消息以来)。

注: BATCHINT 指定等待消息所耗用的总时间。 它不包括检索已在传输队列上可用的消息所花费的时间，也不包括传输消息所花费的时间。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 集群发送方
- 集群接收方

BATCHLIM (批处理限制)

此属性是在获取同步点之前可通过通道发送的数据量的限制 (以千字节为单位)。

同步点是在导致达到限制的消息流过通道后执行的。

该值必须在 0-999999 范围内。 缺省值是 5000。

此属性中的值为零表示没有数据限制应用于此通道上的批处理。

当满足下列其中一个条件时，将结束批处理:

- 已发送 BATCHSZ 消息。
- BATCHLIM 字节已发送。
- 传输队列为空且已超过 BATCHINT。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 集群发送方
- 集群接收方

BATCHSZ (批处理大小)

此属性是在获取同步点之前要发送的最大消息数。

批处理大小不会影响通道传输消息的方式; 消息始终单独传输，但会作为批处理落实或回退。

为了提高性能，您可以设置批处理大小，以定义要在两个 同步点之间传输的最大消息数。 当通道启动时，将协商要使用的批处理大小，并采用两个通道定义中的较低者。 在某些实现上，将根据两个通道定义中的最低值和两个队列管理器 MAXUMSGS 值来计算批处理大小。 批处理的实际大小可以较小; 例如，当传输队列上没有剩余消息或批处理时间间隔到期时，批处理完成。

批处理大小的较大值会增加吞吐量，但恢复时间会增加，因为有更多消息要回退并再次发送。 缺省 BATCHSZ 为 50 ，建议您先尝试该值。 如果您的通信不可靠，那么可以为 BATCHSZ 选择较低的值，从而更有可能需要恢复。

同步点过程需要唯一的逻辑工作单元标识，以便在每次获取同步点时通过链接进行交换，以协调批处理落实过程。

如果已同步的批处理落实过程中断，那么可能会出现 不确定 情况。 当消息通道启动时，将自动解决不确定情境。 如果此解决方案不成功，那么可能需要使用 RESOLVE 命令进行手动干预。

选择批量大小的数字时的一些注意事项:

- 如果此数目过大，那么链路两端占用的队列空间量将过大。 消息在未落实时占用队列空间，并且在落实之前无法从队列中除去这些消息。
- 如果可能有稳定的消息流，那么可以通过增加批处理大小来提高通道的性能，因为传输相同数量的字节需要更少的确认流。

- 如果消息流特征指示消息间歇性到达，那么具有相对较大的断开连接时间间隔的批处理大小 1 可能会提供更好的性能。
- 该数字可以在范围 1 到 9999 之间。
- 即使快速通道上的非持久消息不等待同步点，它们也会影响批量大小计数。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 接收方
- 请求者
- 集群发送方
- 集群接收方

MQSC 关键字的通道属性 (C)

MQSC 关键字的通道属性的字母列表，以字母 *C* 开头。

CERTLABL (证书标签)

此属性指定通道定义的证书标签。

标签标识将密钥存储库中的哪个个人证书发送到远程同级。 如 数字证书标签中所述定义证书。

仅当远程同级的 IBM MQ 版本完全支持证书标签配置并且通道正在使用 TLS CipherSpec 时，入站通道 (包括 RCVR， RQSTR， CLUSRCVR，未限定的 SERVER 和 SVRCONN 通道) 才会发送已配置的证书。

如果不是这种情况，那么队列管理器 **CERTLABL** 属性将确定发送的证书。 此限制是因为入站通道的证书标签选择机制依赖于并非在所有情况下都受支持的 TLS 协议扩展。 尤其是， Java 客户机和 JMS 客户机不支持必需的协议扩展，并且将只接收由队列管理器 **CERTLABL** 属性配置的证书，而不考虑特定于通道的标签设置。

不合格的服务器通道是指没有设置 "**CONNAME** 字段的通道。

没有任何管理接口允许为 CLUSSDR 通道查询或设置此属性。 您将收到一条 MQRCCF_WRONG_CHANNEL_TYPE 消息。 但是，此属性存在于 CLUSSDR 通道对象 (包括 MQCD 结构) 中，如果需要， CHAD 出口可以通过程序设置此属性。

有关证书标签可包含的内容的更多信息，请参阅 数字证书标签，了解需求。

此属性对所有通道类型均有效。

注: 对于 SSL/TLS，必须在 QMGR 定义中定义 "**CERTLABL**。 您可以选择在 **CHANNEL** 定义上设置 **CERTLABL** 。

队列管理器 **CERTLABL** 必须经过验证，且必须为有效的个人证书，即使您在频道定义中设置了 **CERTLABL** 。

CHANNEL (通道名称)

此属性指定通道定义的名称。

该名称最多可以包含 20 个字符，但由于消息通道的两端必须具有相同的名称，并且其他实现可能对大小有限制，因此实际字符数可能必须较小。

在可能的情况下，通道名称对于互连队列管理器网络中任意两个队列管理器之间的一个通道是唯一的。

该名称必须包含以下列表中的字符:

字母顺序	(A-Z， a-z; 注意大写和小写都很重要)
数字	(0-9)
时期	(.)

正斜杠	(/)
下划线	(_)
百分比符号	(%)

注:

1. 不允许嵌入空白，将忽略前导空白。
2. 在使用 EBCDIC 片假名的系统上，不能使用小写字符。

此属性对所有通道类型均有效。

CHLTYPE (通道类型)

此属性指定要定义的通道的类型。

可能的通道类型为:

消息通道类型:

- 发送方
- 服务器
- 接收方
- 请求者
- 集群发送方
- 集群接收方

MQI 通道类型:

- ALW 客户端连接（仅限 AIX, Linux, and Windows

  注: z/OS 还可以在 z/OS 上定义客户机连接通道，以便在其他平台上使用。
- 服务器连接
- AMQP

通道的两端必须具有相同的名称和兼容类型:

- 具有接收方的发送方
- 具有服务器的请求者
- 具有发送方的请求者 (用于回调)
- 具有接收方的服务器 (服务器用作发送方)
- 客户机连接与服务器连接
- 具有集群接收方的集群发送方
- 具有 AMQP 的 AMQP

CLNTWGHT (客户机通道权重)

此属性指定用于影响所使用的客户机连接通道定义的权重。

使用客户机通道权重属性的目的在于当存在多个合适的客户机通道定义时，可以根据其权重随机地选择这些定义。

当客户机通过指定以星号开头的队列管理器名称（可在多个队列管理器之间实现客户机权重平衡）发出请求连接到队列管理器组的 MQCONN 时，如果客户机通道定义表 (CCDT) 中有多个合适的通道定义，则会根据权重随机选择要使用的定义，并按字母顺序首先选择任何适用的 "**CLNTWGHT(0)**定义。

注: 使用 JSON CCDT 时，可以具有多个同名通道。 如果存在多个同名通道，且它们都有 "**CLNTWGHT(0)**，那么将按照 JSON CCDT 中定义的顺序选择通道。

请指定 0 至 99 范围内的值。 缺省值是 0。

零值指示不执行负载均衡并且按字母顺序选择适用的定义。 要启用负载均衡，请指定一个在范围 1 至 99 之间的值，其中 1 是最小的权重，99 是最大的权重。 具有非零权重的两个或多个通道之间的连接分布与这些权重的比率成正比。 例如，"**CLNTWGHT** 值分别为 2、4 和 14 的三个通道被选择的比例分别为 10%、20% 和 70%。 不保证此分布。 如果连接的 "**AFFINITY** 属性设置为 PREFERRED，则第一个连接将根据客户端权重选择通道定义，随后的连接将继续使用相同的通道定义。

此属性仅对客户机连接通道类型有效。

CLUSNL (集群名称列表)

此属性是名称列表的名称，用于指定通道所属的集群的列表。

CLUSTER 或 "**CLUSNL** 的结果值最多只能有一个是非空值。 如果其中一个值为非空白，那么另一个值必须为空白。

此属性对以下通道类型有效:

- 集群发送方
- 集群接收方

集群 (CLUSTER)

此属性是通道所属的集群的名称。

最大长度为 48 个字符，符合 IBM MQ 对象的命名规则。

CLUSTER 或 "**CLUSNL** 的结果值最多只能有一个是非空值。 如果其中一个值为非空白，那么另一个值必须为空白。

此属性仅对以下通道类型有效:

- 集群发送方
- 集群接收方

CLWLPRTY (集群工作负载优先级)

CLWLPRTY 通道属性指定了群集工作负载分配通道的优先级顺序。 该值必须在 0-9 范围内，其中 0 是最低优先级， 9 是最高优先级。

使用 "**CLWLPRTY** 通道属性为可用的群集目的地设置优先顺序。 IBM MQ 先选择优先级最高的目的地，然后再选择集群目的地优先级最低的目的地。 如果存在多个具有相同优先级的目标，那么它将选择最近最少使用的目标。

如果有两个可能的目标，那么可以使用此属性来允许故障转移。 消息转至具有最高优先级通道的队列管理器。 如果它变为不可用，那么消息将转至下一个优先级最高的队列管理器。 优先级较低的队列管理器充当保留。

IBM MQ 先检查通道状态，然后再确定通道的优先级。 只有可用的队列管理器才是可供选择的候选者。

注意:

- 在目标队列管理器上的集群接收方通道上指定此属性。 您在匹配的集群发送方通道上指定的任何均衡都可能被忽略。 请参阅 集群通道。
- 远程队列管理器的可用性取决于该队列管理器的通道状态。 当通道启动时，它们的状态会发生多次更改，其中一些状态对集群工作负载管理算法的优先级较低。 在实践中，这意味着可以在启动高优先级 (主) 目标的通道时选择低优先级 (备份) 目标。
- 如果需要确保没有信息发送到备份目的地，请不要使用 "**CLWLPRTY**。 考虑使用独立队列或 "**CLWLRANK**，从主队列手动切换到备用队列。

CLWLRANK (集群工作负载列组)

CLWLRANK 通道属性指定集群工作负载分布的通道的排名。 该值必须在 0-9 范围内，其中 0 是最低等级， 9 是最高等级。

如果要控制发送到另一个集群中的队列管理器的消息的最终目标，请使用 **CLWLRANK** 通道属性。 通过设置将队列管理器连接到集群交集处的网关队列管理器的通道的列组来控制最终目标的选择。

设置 **CLWLRANK** 时，消息将通过相互连接的集群采用指定的路由，以到达排名较高的目标。 例如，消息到达网关队列管理器，该网关队列管理器可以使用排名 1 和 2 的通道将它们发送到两个队列管理器中的任何一个。 它们将自动发送到由具有最高列组的通道连接的队列管理器，在这种情况下，会将该通道发送到排名 2 的队列管理器。

IBM MQ 在检查通道状态之前获取通道的排名。 在检查通道状态之前获取列组意味着即使不可访问的通道也可供选择。 它允许通过网络路由消息，即使最终目标不可用也是如此。

注意:

- 在目标队列管理器上的集群接收方通道上指定此属性。 您在匹配的集群发送方通道上指定的任何均衡都可能被忽略。 请参阅 集群通道。
- 如果还使用了优先级属性 **CLWLPRTY**，那么 IBM MQ 将在可用目标之间进行选择。 如果通道不可用于具有最高列组的目标，那么消息将保留在传输队列上。 当通道变为可用时，将释放此消息。 消息不会按列组顺序发送到下一个可用目标。

CLWLWGHT (集群工作负载权重)

CLWLWGHT 通道属性指定应用于 CLUSSDR 和 CLUSRCVR 通道的权重，用于群集工作负载分配。 该值必须在 1-99 范围内，其中 1 是最低权重， 99 是最高权重。

使用 "**CLWLWGHT** 可向处理能力更强的服务器发送更多信息。 通道权重越高，通过该通道发送的消息越多。

注意:

- 在目标队列管理器上的集群接收方通道上指定此属性。 您在匹配的集群发送方通道上指定的任何均衡都可能被忽略。 请参阅 集群通道。
- 当 "**CLWLWGHT** 从任何通道的默认值 50 变更时，工作量平衡将取决于发送到任何群集队列的报文选择每个通道的总次数。 有关更多信息，请参阅 第 126 页的『集群工作负载管理算法』。

COMPHDR (头压缩)

此属性是通道支持的头数据压缩技术的列表。

对于发送方，服务器，集群发送方，集群接收方和客户机连接通道，指定的值与正在使用的通道的远程端支持的第一种压缩技术按优先顺序排列。 将通道的相互支持的压缩技术传递到发送通道的消息出口，在该出口中，可以根据每条消息来更改所使用的压缩技术。 压缩会改变传递到发送和接收出口的数据。

可能的值为:

NONE
不执行头数据压缩。 该值为缺省值。

SYSTEM
执行头数据压缩。

此属性对所有通道类型均有效。

COMPMSG (数据压缩)

此属性是通道支持的消息数据压缩技术的列表。

对于发送方，服务器，集群发送方，集群接收方和客户机连接通道，指定的值按优先顺序排列。 使用通道的远程端支持的第一种压缩技术。 将通道的相互支持的压缩技术传递到发送通道的消息出口，在该出口中，可以根据每条消息来更改所使用的压缩技术。 压缩会改变传递到发送和接收出口的数据。 请参阅 第 88 页的『COMPHDR (头压缩)』 以获取消息头的压缩。

可能的值为：

NONE
不执行消息数据压缩。 该值为缺省值。

RLE
使用运行长度编码执行消息数据压缩。

ZLIBFAST
使用 zlib 压缩技术来执行消息数据压缩。 推荐使用快速压缩时间。

可以选择将 ZLIBFAST 卸载到 zEnterprise 数据压缩工具。 请参阅 zEDC Express 设施 以获取更多信息。

ZLIBHIGH
使用 zlib 压缩技术来执行消息数据压缩。 推荐使用高级压缩。

V 9.4.0 Multi **LZ4FAST**
使用 LZ4 压缩技术执行消息数据压缩。 推荐使用快速压缩时间。

V 9.4.0 Multi **LZ4HIGH**
消息数据压缩是使用 LZ4 压缩技术执行的。 推荐使用高级压缩。

ANY
允许通道支持队列管理器支持的任何压缩技术。 仅支持 "接收方"， "请求者" 和 "服务器连接" 通道。

此属性对所有通道类型均有效。

AIX 从 IBM MQ 9.3.0 开始， ZLIBFAST 和 ZLIBHIGH 技术可以在 IBM MQ for AIX 上使用硬件加速的 zlibNX 库 (如果已安装)。 zlibNX 库是 zlib 压缩库的增强版本，通过在基于 IBM POWER9 处理器的服务器上使用称为 Nest 加速器 (NX) 的协处理器，支持硬件加速的数据压缩和解压。 zlibNX 库在 IBM AIX 7.2 with Technology Level 4 Expansion Pack 和更高版本中提供。 大小超过 2KB 的高度可压缩消息最有可能从选择使用 zlibNX 库 (通过减少 CPU 使用率) 中获益。 要使消息通道代理程序 (MCA) 能够使用 zlibNX 库，请设置环境变量 `AMQ_USE_ZLIBNX`。

CONNAME (连接名称)

此属性是通信连接标识。 它指定此通道使用的特别通信链路。

它对于服务器通道是可选的，除非触发了服务器通道，在这种情况下，它必须指定连接名称。

将 **CONNAME** 指定为以逗号分隔的机器（用于声明的 **TRPTYPE**）的名称列表。 通常只需要一个机器名称。 您可以提供多个机器名称来配置具有相同属性的多个连接。 通常会采用连接列表中指定的顺序尝试连接，直至成功建立连接。 如果提供了 **CLNTWGHT** 属性，那么将针对客户机修改顺序。 如果没有任何连接成功，那么通道会重试连接，具体由该通道的属性确定。 通过客户机通道，连接列表提供了除使用队列管理器组之外的替代方法来配置多个连接。 通过消息通道，连接列表用于配置与多实例队列管理器的替代地址的连接。

Multi 在 多平台 上，集群接收方通道的 TCP/IP 连接名称参数是可选的。 如果您将连接名称留空，IBM MQ 将为您生成一个连接名称，采用缺省端口并使用系统的当前 IP 地址。 您可以覆盖缺省端口号，但仍使用系统的当前 IP 地址。 对于每个连接名称，请将 IP 名称留空，并在括号内提供端口号；例如：

```
(1415)
```

生成的 **CONNAME** 始终采用点分十进制 (IPv4) 或十六进制 (IPv6) 格式，而不采用字母数字 DNS 主机名形式。

Multi 在 "多平台中，字符串的最大长度为 264 个字符。

z/OS 在 "z/OS 中，"**CONNAME** 的最大长度各不相同：

- 对于 "**TRPTYPE(LU62)** 频道，最大长度始终为 48 个字符。
- LTS 对于 **TRPTYPE(TCP)**、 Long Term Support 和 Continuous Delivery （在 IBM MQ 9.4.3 之前）的频道，最大长度为 48 个字符。

- V 9.4.3 对于 **TRPTYPE(TCP)** 的频道，从 IBM MQ 9.4.3 开始，最大长度为 264 个字符。 在队列共享组中的所有队列管理器都访问 IBM MQ 9.4.3 或更高版本之前，长度大于 48 个字符的 **QSGDISP(GROUP|COPY)** 通道定义不能使用。
- 对于客户端连接通道，最大长度始终为 48 个字符。

对于 **TRPTYPE(TCP)** 的频道，以下建议可能是解决 48 个字符限制的办法之一：

- V 9.4.3 迁移到 "IBM MQ 9.4.3 或更高版本。
- 设置 DNS 服务器，例如使用 `myserver` 而不是 `myserver.location.company.com` 作为主机名，确保可以使用简短的主机名。
- 使用 IP 地址。

如果传输类型为 TCP

CONNAME 是远程机器（或群集接收器通道的本地机器）的主机名或网络地址。 例如，`"(ABC.EXAMPLE.COM)`、`"(2001:DB8:0:0:0:0:0:0)`或 `"(127.0.0.1)`。 它可以包括端口号，例如 `"(MACHINE(123))`。

z/OS 它可以包含动态 DNS 组或 Network Dispatcher 输入端口的 IP_name。

如果在仅支持 IPv4 的网络中使用 IPv6 地址，那么不会解析连接名称。 在同时使用 IPv4 和 IPv6 的网络中，连接名称与本地地址进行交互以确定使用的 IP 堆栈。

有关更多信息，请参阅第 93 页的『LOCLADDR (本地地址)』。

如果传输类型为 LU 6.2

Multi 如果指定了 "**TPNAME** 和 "**MODENAME**，则给出伙伴 LU 的全称。 如果 "**TPNAME** 和 "**MODENAME** 为空，请给出特定平台的 CPI-C 侧信息对象名称。

z/OS 有两种格式可用于指定值：

- 逻辑单元名称

  队列管理器的逻辑单元信息包括逻辑单元名、TP 名和可选方式名。 可以通过以下三种格式之一指定此名称：

  表 *55:* 逻辑单元名称和格式

表单	**示例**
luname	IGY12355
luname/TPname	IGY12345/APING
luname/TPname/modename	IGY12345/APINGD/#INTER

  对于第一种表格，必须为 "**TPNAME** 和 "**MODENAME** 属性指定 TP 名称和模式名称，否则这些属性必须为空。 对于客户机连接通道，只允许第一种形式。

- 符号名称

  队列管理器的逻辑单元信息的符号目标名称，在辅助信息数据集中定义。 **TPNAME** 和 "**MODENAME** 属性必须为空。 请注意，对于集群接收方通道，辅助信息位于集群中的其他队列管理器上。 在这种情况下，它可以是通道自动定义出口可以解析为本地队列管理器的相应逻辑单元信息的名称。

指定或隐含的 LU 名可以是 VTAM 通用资源组的 LU 名。

如果传输协议为 NetBIOS

CONNAME 是远程计算机上定义的 NetBIOS 名称。

如果传输协议是 SPX

CONNAME 是 SPX 类型的地址，由 4 字节网络地址、6 字节节点地址和 2 字节套接字编号组成。 以十六进制输入这些值，网络和节点地址之间用句点和方括号中的套接字号分隔。 例如：

```
CONNAME('0a0b0c0d.804abcde23a1(5e86)')
```

如果省略了套接字号，那么将使用缺省 IBM MQ SPX 套接字号。 默认值为 X'5E86'。

此属性对以下通道类型有效:

- 发送方
- 服务器

  它对于服务器通道是可选的，除非触发了服务器通道，在这种情况下，它必须指定连接名称。
- 请求者
- 客户机连接
- 集群发送方
- 集群接收方

注: 传输协议的定义包含在 TRPTYPE 中。

CONVERT (转换消息)

此属性指定在传输之前必须将消息转换为接收系统所需的格式。

应用程序消息数据通常由接收应用程序转换。 但是，如果远程队列管理器位于不支持数据转换的平台上，请使用此通道属性指定必须**在传输前** 将消息转换为接收系统所需的格式。

可能的值为 yes 和 no。 如果指定 "**是**"，则在发送前转换报文中的应用程序数据（如果已指定内置格式名称，或用户定义的格式有数据转换出口）（请参阅 "编写数据转换出口"）。 如果指定 no，那么在发送之前不会转换消息中的应用程序数据。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 集群发送方
- 集群接收方

MQSC 关键字 (D-L) 的通道属性

MQSC 关键字的通道属性的字母列表，以字母 *D* 到 *L* 开头。

DEFRECON (缺省重新连接)

指定客户机连接是否在其连接中断时自动重新连接客户机应用程序。

可能的值为:

NO (缺省值)
除非被 **MQCONNX** 覆盖，否则客户端不会自动重新连接。

YES
除非被 **MQCONNX** 覆盖，否则客户端会自动重新连接。

QMGR
除非被 **MQCONNX** 覆盖，否则客户端会自动重新连接，但只能连接到同一个队列管理器。 选项 QMGR 的效果与 MQCNO_RECONNECT_Q_MGR 相同。

DISABLED
即使客户程序使用 **MQCONNX**MQI 调用请求重新连接，也会被禁用。

此属性仅对客户机连接通道有效。

DESCR (描述)

此属性描述通道定义，最多包含 64 个字节的文本。

注: 如果系统使用双字节字符集(DBCS),那么最大字符数将减少。

使用由队列管理器的编码字符集标识 (CCSID) 标识的字符集中的字符，以确保文本在发送到另一个队列管理器时正确转换。

此属性对所有通道类型均有效。

DISCINT (断开连接时间间隔)

此属性是通道关闭的时间长度 (如果在该时间段内没有消息到达)。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 仅使用 TCP 协议的服务器连接
- 集群发送方
- 集群接收方

此属性是超时，以秒为单位指定。

对于服务器到服务器消息通道 (服务器，集群发送方，发送方和集群接收方通道类型)，时间间隔从批处理结束点 (即达到批处理大小或批处理时间间隔到期且传输队列变为空时) 开始测量。 如果在指定的时间间隔内没有消息到达传输队列，那么通道将关闭。 (时间大致。)

在服务器到服务器消息通道的两端之间的控制数据的关闭交换包括关闭原因的指示。 这将确保通道的相应端保持可用状态以再次启动。

您可以指定从 0 到 999 999 之间的任何秒数，其中值为 0 表示不断开连接; 请无限期等待。

对于服务器到服务器消息通道，缺省值为 6000 秒 (100 分钟)，对于服务器连接通道，缺省值为 0 (无超时)。 您可以通过变更缺省通道对象来变更用于您创建的新通道的缺省值。 例如，更改 SYSTEM.DEF.SENDER 上的 **DISCINT** 属性，为您定义的新发件人信道提供新的默认值。

对于使用 TCP 协议的服务器连接通道，时间间隔表示客户机不活动断开连接值 (以秒为单位指定)。 如果服务器连接通道程序在此持续时间内未接收到来自其伙伴客户机的通信，那么它将终止该连接。

服务器连接不活动时间间隔适用于来自客户机的 IBM MQ API 调用。

注: 可能长期运行的带等待调用 MQGET 不会被归类为不活动，因此不会因 **DISCINT** 过期而超时。

注意: 性能受为断开连接时间间隔指定的值影响。

低值 (例如，几秒) 可能通过不断停止和重新启动通道来损害系统性能。 较大的值 (超过一个小时) 可能意味着系统资源将被耗用而无益。 您还可以指定脉动信号间隔，以便当传输队列上没有消息时，发送 MCA 向接收 MCA 发送脉动信号流，从而使接收 MCA 有机会停顿通道，而无需等待断开连接间隔到期。 要使这两个值有效协同工作，脉动信号间隔值必须显着低于断开连接间隔值。

服务器到服务器报文通道的 **DISCINT** 默认值为 6000 秒（100 分钟）。 但是，几分钟的值通常是一个合理的值，可以在不影响性能或使通道长时间运行的情况下使用。 如果适合您的环境，您可以在创建自己的通道之前，在每个单独通道上或通过更改默认通道定义中的 **DISCINT** 属性（例如，发送者通道的 SYSTEM.DEF.SENDER ）来更改该值。

有关更多信息，请参阅 停止和停顿通道。

HBINT (脉动信号间隔)

此属性指定当传输队列上没有消息时，要从发送消息通道代理程序 (MCA) 传递的脉动信号流之间的近似时间。

脉动信号流不会阻塞在接收的 MCA，该 MCA 正在等待消息到达或等待断开连接的时间间隔期满。 当取消阻塞接收 MCA 后，它可以将通道断开连接，而不需要等待断开连接时间间隔期满。 脉动信号流还释放为大型消息分配的任何存储缓冲区，并关闭通道接收端保持打开的任何队列。

该值以秒为单位，必须在 0-999 999 范围内。 值为零表示不发送脉动信号流。 缺省值为 300。 为了最有用，该值必须显着小于断开连接时间间隔值。

对于使用 IBM MQ classes for Java， JMS 或 .NET API 的应用程序，将通过下列其中一种方法来确定 HBINT 值:

- 通过应用程序使用的 SVRCONN 通道上的值。
- 或者按 CLNTCONN 通道上的值 (如果应用程序已配置为使用 CCDT)。

对于服务器连接和客户机连接通道，脉动信号可以分别从服务器端和客户端流动。 如果在脉动信号间隔内未通过通道传输任何数据，那么客户机连接 MQI 代理程序将发送脉动信号流，而服务器连接 MQI 代理程序将使用另一个脉动信号流对其进行响应。 发生这种情况与通道的状态无关，例如，无论它是在进行 API 调用时处于不活动状态，还是在等待客户机用户输入时处于不活动状态。 无论通道的状态如何，服务器连接 MQI 代理程序也能够再次向客户机启动脉动信号。 为了防止服务器连接和客户机连接 MQI 代理程序同时相互跳动，在脉动信号间隔加 5 秒内，在通道中未传输任何数据后，将会流动服务器脉动信号。

对于在 IBM WebSphere MQ 7.0 之前的通道方式下工作的服务器连接和客户机连接通道，仅当服务器 MCA 正在等待指定了 WAIT 选项的 MQGET 命令 (它已代表客户机应用程序发出) 时，脉动信号才会流动。

有关使 MQI 通道在两种方式下工作的更多信息，请参阅 SharingConversations (MQLONG)。

KAINT (保持活动时间间隔)

此属性用于指定通道的超时值。

"保持活动时间间隔" 属性是传递到通信堆栈的值，用于指定通道的保持活动计时。 它允许您为每个通道指定不同的保持活动值。

您可以为每个通道设置 Keepalive Interval (**KAINT**) 属性。

Multi 在 多平台上，您可以访问和修改参数，但仅存储和转发该参数; 没有该参数的功能实现。 如果需要 **KAINT** 参数提供的功能，请使用心跳间隔 (**HBINT**) 参数，如 第 92 页的『HBINT (脉动信号间隔)』 所述。

要使此属性具有任何效果，必须启用 TCP/IP 保持活动。

- z/OS 在 z/OS 上，可通过发布 **ALTER QMGR TCPKEEP(YES)** MQSC 命令启用 keepalive。
- Multi 在 多平台 上，如果在分布式队列配置文件 `qm.ini` 的 TCP 节中指定了 **KEEPALIVE=YES** 参数，或通过 IBM MQ Explorer 指定了 参数，就会出现这种情况。

  还必须使用 TCP 概要文件配置数据集在 TCP/IP 本身中启用保持活动。

该值指示时间 (以秒计)，并且必须在范围 0-99999 内。 "保持活动时间间隔" 值 0 指示未对通道启用特定于通道的保持活动，仅使用 TCP/IP 中设置的系统范围保持活动值。 您也可以将 **KAINT** 设置为**自动** （默认值）。 如果 **KAINT** 设置为 AUTO，则 Keepalive 值基于协商的心跳间隔值 (**HBINT**) 如下:

表 *56:* 协商的 *HBINT* 值和相应的 *KAINT* 值

协商的 HBINT	**KAINT**
>0	协商的 HBINT + 60 秒
0	0

此属性对所有通道类型均有效。

对于 TransportType (**TRPTYPE**) 而非 TCP 或 SPX 的所有通道，该值将被忽略。

LOCLADDR (本地地址)

此属性指定通道的本地通信地址。

注: AMQP 信道不支持与其他 IBM MQ 信道相同的 **LOCLADDR** 格式。 更多信息，请参阅 AMQP 信道的 LOCLADDR。

所有通道 (AMQP 通道除外) 的 LOCLADDR

仅当传输类型 (**TRPTYPE**) 为 TCP/IP 时，此属性才适用。 对于所有其他传输类型，将忽略该传输类型。

当指定 **LOCLADDR** 值时，停止后重新启动的通道将继续使用 LOCLADDR 中指定的 TCP/IP 地址。 在恢复方案中，当通道通过防火墙进行通信时，此属性可能很有用。 这很有用，因为它将除去由于通道重新启动而导致的问题 (使用它所连接的 TCP/IP 堆栈的 IP 地址)。 **LOCLADDR** 还可以强制通道在双协议栈系统上使用 或 协议栈，或在单协议栈系统上使用双模协议栈。 IPv4 IPv6

此属性对以下通道类型有效:

- 发送方
- 服务器
- 请求者
- 客户机连接
- 集群发送方
- 集群接收方

当 **LOCLADDR** 包含网络地址时，该地址必须是属于运行通道的系统上的网络接口的网络地址。 例如，使用以下 MSQC 命令在队列管理器 ALPHA 上定义发送方通道到队列管理器 BETA 时:

```
DEFINE CHANNEL(TO.BETA) CHLTYPE(SDR) CONNAME(192.0.2.0) XMITQ(BETA) LOCLADDR(192.0.2.1)
```

LOCLADDR 地址是 IPv4 地址 192.0.2.1。 此发送方通道在队列管理器 ALPHA 的系统上运行，因此 IPv4 地址必须属于其系统的其中一个网络接口。

该值是可选的 IP 地址，以及用于出站 TCP/IP 通信的可选端口或端口范围。 此信息的格式如下所示:

```
LOCLADDR([ip-addr][(low-port[,high-port])][,[ip-addr][(low-port[,high-port])]])
```

LOCLADDR 的最大长度（包括多个地址）为 MQ_LOCAL_ADDRESS_LENGTH。

如果省略 **LOCLADDR**，那么将自动分配本地地址。

请注意，您可以使用客户机通道定义表 (CCDT) 为 C 客户机设置 **LOCLADDR**。

所有参数都是可选的。 省略地址的 ip-addr 部分有助于为 IP 防火墙启用固定端口号配置。 省略端口号有助于选择特定网络适配器，而无需识别唯一的本地端口号。 TCP/IP 堆栈将生成唯一的端口号。

对每个附加本地地址多次指定 *[,[ip-addr][(low-port[,high-port])]]*。 如果要指定本地网络适配器的特定子集，请使用多个本地地址。 您还可以使用 *[,[ip-addr][(low-port[,high-port])]]* 来表示多实例队列管理器配置中的不同服务器上的特定本地网络地址。

ip-addr
ip-addr 通过以下三种格式之一来指定:

IPv4 点分十进制
例如，192.0.2.1

IPv6 十六进制表示法
例如，2001:DB8:0:0:0:0:0:0

字母数字主机名格式
例如 WWW.EXAMPLE.COM

low-port 和 high-port
low-port 和 high-port 是括在括号中的端口号。

下表显示 **LOCLADDR** 参数的使用方式:

表 *57:* 有关如何使用 ***LOCLADDR*** 参数的示例

LOCLADDR	**含义**
9.20.4.98	通道在本地绑定到此地址
9.20.4.98, 9.20.4.99	通道绑定到任一 IP 地址。 该地址可能是一个服务器上的两个网络适配器，或者是多实例配置中两个不同服务器上的不同网络适配器。

表 57: 有关如何使用 ***LOCLADDR*** 参数的示例 (继续)	
LOCLADDR	**含义**
9.20.4.98(1000)	通道在本地绑定到此地址和端口 1000
9.20.4.98(1000,2000)	通道在本地绑定到此地址并使用 1000 - 2000 范围内的端口
(1000)	通道在本地绑定到端口 1000
(1000,2000)	通道在本地绑定到 1000 - 2000 范围内的端口

当通道启动时，为连接名称 (**CONNAME**) 和本地地址 (**LOCLADDR**) 指定的值确定用于通信的 IP 堆栈。 使用的 IP 堆栈确定如下:

- 如果系统仅配置了 IPv4 堆栈，那么将始终使用 IPv4 堆栈。 如果将本地地址 (**LOCLADDR**) 或连接名称 (**CONNAME**) 指定为 IPv6 网络地址，那么将生成错误并且通道无法启动。
- 如果系统仅配置了 IPv6 堆栈，那么将始终使用 IPv6 堆栈。 如果将本地地址 (**LOCLADDR**) 指定为 IPv4 网络地址，那么将生成错误并且通道无法启动。 在支持 IPv6 映射寻址的平台上，如果连接名称 (**CONNAME**) 被指定为 IPv4 网络地址，则该地址会被映射为 IPv6 地址。 例如， `xxx.xxx.xxx.xxx` 映射到 `::ffff:xxx.xxx.xxx.xxx`。 使用映射地址可能需要协议转换程序。 尽可能避免使用映射的地址。
- 如果将本地地址 (**LOCLADDR**) 指定为通道的 IP 地址，那么将使用该 IP 地址的堆栈。 如果本地地址 (LOCLADDR) 被指定为同时解析到 IPv4 和 IPv6 地址的主机名，则由连接名称 (**CONNAME**) 决定使用哪个堆栈。 如果本地地址 (**LOCLADDR**) 和连接名称 (**CONNAME**) 都指定为同时解析到 IPv4 和 IPv6 地址的主机名，则使用的堆栈由队列管理器属性 **IPADDRV** 决定。
- 如果系统配置了双 IPv4 和 IPv6 堆栈，并且没有为通道指定本地地址 (**LOCLADDR**) ，那么为该通道指定的连接名称 (**CONNAME**) 将确定要使用的 IP 堆栈。 如果连接名 (**CONNAME**) 指定为同时解析到 IPv4 和 IPv6 地址的主机名，则使用的堆栈由队列管理器属性 **IPADDRV** 决定。

Multi 在 多平台上，可以设置用于所有未定义本地地址的发送方通道的缺省本地地址值。 默认值通过在启动队列管理器之前设置 **MQ_LCLADDR** 环境变量来定义。 该值的格式与 MQSC 属性 LOCLADDR 的格式一致。

具有集群发送方通道的本地地址

集群发送方通道始终继承目标队列管理器上定义的相应集群接收方通道的配置。 即使存在同名的本地定义集群发送方通道，也是如此，在这种情况下，手动定义仅用于初始通信。

因此，不可能依赖群集接收器信道中定义的 **LOCLADDR** ，因为 IP 地址很可能不属于创建群集发送器的系统。 因此，不应使用群集接收器上的 **LOCLADDR** ，除非有理由只限制端口而不限制所有潜在群集发送者的 IP 地址，而且已知这些端口在可能创建群集发送者通道的所有系统上都可用。

如果群集必须使用 **LOCLADDR** 才能将出站通信通道绑定到特定 IP 地址，则应使用通道自动定义出口 ，或尽可能使用队列管理器的默认 **LOCLADDR** 。 使用通道出口时，会将出口的 **LOCLADDR** 值强制导入任何一个自动定义的 CLUSSDR 通道。

如果通过使用退出或默认值来为群集发送者通道使用非默认的 **LOCLADDR** ，则任何匹配的手动定义的群集发送者通道（例如到完整存储库队列管理器的通道）也必须设置 **LOCLADDR** 值，以启用通道上的初始通信。

注: 如果操作系统对 **LOCLADDR** 中提供的端口 (或所有端口，如果提供了端口范围) 返回绑定错误，那么通道不会启动; 系统会发出错误消息。

针对 AMQP 通道的 LOCLADDR

AMQP 信道支持与其他 IBM MQ 信道不同的 **LOCLADDR** 格式:

LOCLADDR (*ip-addr*)

LOCLADDR 是通道的本地通信地址。 如果要强制客户机使用特定 IP 地址，请使用此参数。 如果选项可用，那么强制通道使用 IPv4 或 IPv6 地址，或者在具有多个网络适配器的系统上使用特定网络适配器，**LOCLADDR** 也很有用。

LOCLADDR 的最大长度为 MQ_LOCAL_ADDRESS_LENGTH。

如果省略 **LOCLADDR**，那么将自动分配本地地址。

ip-addr
ip-addr 是单一网络地址，以三种格式之一指定:

IPv4 点分十进制
例如 192.0.2.1

IPv6 十六进制表示法
例如 2001:DB8:0:0:0:0:0:0

字母数字主机名格式
例如 WWW.EXAMPLE.COM

如果输入了 IP 地址，那么将仅验证地址格式。 未验证 IP 地址本身。

请参阅 使用自动定义的集群发送方通道 以获取更多信息。

LONGRTY (长重试计数)

此属性指定通道尝试将会话分配给其合作伙伴的最大次数。

长重试次数属性 (**LONGRTY**) 的设置范围为 0 至 999 999 999。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 集群发送方
- 集群接收方

如果初始分配尝试失败，那么将递减 *short retry count* 号，并且通道将重试剩余次数。 如果它仍然失败，那么它将重试 *long retry count* 次，每次尝试之间的时间间隔为 *long retry interval* 。 如果仍未成功，那么通道将关闭。 然后，必须使用命令重新启动通道; 通道启动程序不会自动启动该通道。

z/OS 在 z/OS 上，如果已超过最大通道数 (**MAXCHL**) ，那么通道无法进入重试。

Multi 在 多平台上，为了尝试重试，通道启动程序必须正在运行。 通道启动程序必须监视在通道正在使用的传输队列定义中指定的启动队列。

如果在通道重试期间停止了 通道启动程序 (在 z/OS 上) 或 通道 (在 多平台上) ，那么在通道启动程序或通道重新启动时，或者在发送方通道成功放入消息时，将重置 短重试计数 和 长重试计数 。 但是，如果 通道启动程序 (在 z/OS 上) 或者 队列管理器 (在多平台上) 关闭并重新启动，不会重置 短重试计数 和 长重试计数 。 通道保留在队列管理器重新启动或放入消息之前的重试计数值。

Multi 在 多平台 上:

1. 当通道从 RETRYING 状态转为 RUNNING 状态时， 短重试计数和长重试计数不会立即重置。 仅当通道进入 RUNNING 状态后第一条消息成功流过通道时，即当本地通道确认发送到另一端的消息数量时，才会重置它们。
2. 重新启动通道时，将重置 短重试计数 和 长重试计数 。

LONGTMR (长时间重试时间间隔)

此属性是在长时间重试方式下，通道在重试建立连接之前要等待的大致时间间隔 (以秒为单位)。

如果通道必须等待变为活动状态，那么可以延长重试之间的时间间隔。

在此长时间间隔内，通道尝试连接 *long retry count* 次，在短重试时间间隔内尝试 *short retry count* 次。

此属性可以设置为从零到 999 999 999。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 集群发送方
- 集群接收方

MQSC 关键字的通道属性 (M)

MQSC 关键字的通道属性的字母顺序列表，以字母 *M* 开头。

MAXINST (最大实例数)

此属性指定可以启动的服务器连接通道或 AMQP 通道的最大并发实例数。

服务器连接通道连接的最大实例数

对于服务器连接通道，此属性指定可以启动的服务器连接通道的最大并发实例数。

可以将此属性设置为从 0 到 999 999 999。 值为零时表示在该通道上不允许进行任何客户机连接。 缺省值为 999 999 999。

如果减小该值以使其小于当前正在运行的服务器连接通道的实例数，那么不会影响正在运行的通道。 但是，在已有足够的实例停止运行之前，新实例无法启动。

AMQP 通道连接的最大实例数

对于 AMQP 通道，此属性指定可启动的 AMQP 通道的最大并发实例数。

可以将此属性设置为从 0 到 999 999 999。 值为零时表示在该通道上不允许进行任何客户机连接。 缺省值为 999 999 999。

如果客户机尝试连接，并且已连接的客户机数已达到 MAXINST，那么通道将关闭具有关闭帧的连接。 关闭框架包含以下消息:

```
amqp:resource-limit-exceeded
```

如果客户机使用已连接的标识进行连接 (即，它执行客户机接管)，那么无论已连接的客户机数是否已达到 MAXINST，接管都将成功。

请参阅 服务器连接通道限制 以获取其他信息。

MAXINSTC (每个客户机的最大实例数)

此属性指定可以从单个客户机启动的服务器连接通道的最大并发实例数。

可以将此属性设置为从 0 到 999 999 999。 值为零时表示在该通道上不允许进行任何客户机连接。 缺省值为 999 999 999。

如果减小该值以使其小于当前从各个客户机运行的服务器连接通道的实例数，那么正在运行的通道不受影响。 但是，这些客户机的新实例在足够的现有实例停止运行之前无法启动。

此属性仅对于服务器连接通道有效。

请参阅 服务器连接通道限制 以获取其他信息。

MAXMSGL (最大消息长度)

此属性指定可以在通道上传输的消息的最大长度。

Multi 在 多平台上，指定大于或等于零的值，并小于或等于队列管理器的最大消息长度。 请参阅 ALTER QMGR 中的 ALTER QMGR 命令的 MAXMSGL 参数，以获取更多信息。

z/OS 在 IBM MQ for z/OS 上，指定大于或等于零且小于或等于 104 857 600 字节 (即 100 MB) 的值。

由于不同的平台上存在各种 IBM MQ 系统实施，因此在某些应用程序中，可用于消息处理的大小可能受到限制。 此数字必须反映系统能够无压力地处理的大小。 启动通道时，将采用通道每一端的两个数字中较小的一个。

注: 您可以对通道使用最大消息大小 0 ，这表示该大小将设置为本地队列管理器最大值。

通过将数字签名和密钥添加到消息中， Advanced Message Security 会增加消息长度。

此属性对所有通道类型均有效。

MCANAME (消息通道代理程序名称)

此属性是保留的，如果指定此属性，那么必须仅设置为空白，并且最大长度为 20 个字符。

MCATYPE (消息通道代理程序类型)

此属性可以将消息通道代理程序指定为进程或线程。

作为进程运行的优点包括:

- 每个通道的隔离提供更大的完整性
- 特定于每个通道的作业权限
- 控制作业调度

线程的优点包括:

- 大大减少了对存储的使用
- 通过在命令行上输入更简单的配置
- 执行速度更快-启动线程比指示操作系统启动进程更快

注: 对于发送方，服务器和请求者的通道类型，缺省值为 `process`。 对于集群发送方和集群接收方的通道类型，缺省值为 `thread`。 在安装期间，可以更改这些缺省值。

如果在通道定义上指定 `process` ，那么将启动 RUNMQCHL 进程。 如果指定 `thread`，那么 MCA 将在 AMQRMPPA 进程的线程上运行，如果指定了 **MQNOREMPOOL** ，那么将在 RUNMQCHI 进程的线程上运行。 在接收入站分配的机器上，如果使用 **RUNMQLSR**，那么 MCA 将作为线程运行。 如果使用 **inetd**，那么它将作为进程运行。

z/OS 在 IBM MQ for z/OS 上，只有通道类型为集群接收方的通道才支持此属性。

Multi 在其他平台上，此属性对以下通道类型有效:

- 发送方
- 服务器
- 请求者
- 集群发送方
- 集群接收方

MCAUSER (消息通道代理程序用户标识)

此属性是 MCA 用于授权访问 IBM MQ 资源的用户标识 (字符串)。

注: 为要运行的通道提供用户标识的替代方法是使用通道认证记录。 通过通道认证记录，不同的连接可以使用相同的通道，同时使用不同的凭证。 如果设置了通道上的两个 MCAUSER ，并且通道认证记录用于应用于同一通道，那么通道认证记录优先。 仅当通道认证记录使用 USERSRC (CHANNEL) 时，才会使用通道定义上的 MCAUSER。

此授权包括 (如果 PUT 权限为 DEF) 将消息放入接收方或请求者通道的目标队列。

在 IBM MQ for Windows 上，可以使用格式 user@domain 对用户标识进行域限定，其中 domain 必须是本地系统的 Windows 系统域或可信域。

如果此属性为空，那么 MCA 将使用其缺省用户标识。 有关更多信息，请参阅 DEFINE CHANNEL。

此属性对以下通道类型有效:

- 接收方
- 请求者
- 服务器连接
- 集群接收方

MODENAME (LU 6.2 方式名)

此属性用于 LU 6.2 连接。 执行通信会话分配时，此属性为连接的会话特征提供额外定义。

当对 SNA 通信使用辅助信息时，方式名在 CPI-C 通信辅助对象或 APPC 辅助信息中定义，并且此属性必须留空; 否则，必须将其设置为 SNA 方式名。

名称长度必须为 1 到 8 个字母数字字符。

此属性仅对以下通道类型有效:

- 发送方
- 服务器
- 请求者
- 客户机连接
- 集群发送方
- 集群接收方

MONCHL (监视)

此属性控制联机监视数据的收集。

可能的值为:

QMGR
联机监视数据的集合继承自队列管理器对象中 MONCHL 属性的设置。 该值为缺省值。

OFF
已禁用此通道的联机监视数据收集。

低
数据收集比率低，对性能的影响最小。 但是，显示的监视结果可能不是最新的。

中型
数据收集的适度比率，对系统性能的影响有限。

高
数据收集的高比率，可能对性能产生影响。 但是，显示的监视结果是最新的。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 接收方
- 请求者
- 服务器连接
- 集群发送方
- 集群接收方

有关监视数据的更多信息，请参阅 显示队列和通道监视数据。

MRDATA (消息重试出口用户数据)

其属性指定在调用时传递到通道消息重试出口的数据。

此属性对以下通道类型有效:

- 接收方
- 请求者
- 集群接收方

MREXIT (消息重试出口名称)

此属性指定要由消息重试用户出口运行的用户出口程序的名称。

如果没有消息重试出口程序生效，请留空。

名称的格式和最大长度取决于平台，如 第 104 页的『RCVEXIT (接收出口名称)』。 但是，只能指定一个消息重试出口。

此属性对以下通道类型有效:

- 接收方
- 请求者
- 集群接收方

MRRTY (消息重试计数)

此属性指定通道尝试重新传递消息的次数。

仅当消息重试出口名称为空时，此属性才控制 MCA 的操作。 如果出口名称不为空，那么会将 MRRTY 的值传递到该出口，但所进行的尝试次数 (如果有) 由该出口控制，而不是由该属性控制。

该值必须在范围 0-999 999 999 之间。 值为零表示不进行任何其他尝试。 缺省值是 10。

此属性对以下通道类型有效:

- 接收方
- 请求者
- 集群接收方

MRTMR (消息重试时间间隔)

此属性指定通道可以重试 MQPUT 操作之前必须经过的最短时间间隔 (以毫秒为单位)。

仅当消息重试出口名称为空时，此属性才控制 MCA 的操作。 如果出口名称不为空，那么 MRTMR 的值将传递到出口以供出口使用，但重试时间间隔由出口控制，而不是由此属性控制。

该值必须在范围 0-999 999 999 之间。 值为零表示尽快执行重试 (如果 MRRTY 的值大于零)。 缺省值是 1000。

此属性对以下通道类型有效:

- 接收方
- 请求者
- 集群接收方

MSGDATA (消息出口用户数据)

此属性指定传递到通道消息出口的用户数据。

您可以运行一系列消息出口。 对用户数据长度的限制以及如何为多个出口指定 MSGDATA 的示例如 RCVDATA 所示。 请参阅 第 104 页的『RCVDATA (接收出口用户数据)』。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 接收方
- 请求者
- 集群发送方
- 集群接收方

MSGEXIT (消息出口名称)

此属性指定要由通道消息出口运行的用户出口程序的名称。

此属性可以是要连续运行的程序的名称列表。 如果没有通道消息出口生效，请留空。

此属性的格式和最大长度取决于平台，如 第 104 页的『RCVEXIT (接收出口名称)』。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 接收方
- 请求者
- 集群发送方
- 集群接收方

MQSC 关键字的通道属性 (N-R)

MQSC 关键字的通道属性的字母顺序列表，以字母 *N* 到 *R* 开头。

NETPRTY (网络连接优先级)

NETPRTY 通道属性指定 CLUSRCVR 通道的优先级。 该值必须在 0-9 范围内，其中 0 是最低优先级， 9 是最高优先级。

使用 NETPRTY 属性使一个网络成为主网络，另一个网络成为备份网络。 给定一组排名相同的通道，集群会在多个路径可用时选择具有最高优先级的路径。

使用 NETPRTY 通道属性的典型示例是区分具有不同成本或速度并连接相同目标的网络。

注: 在目标队列管理器上的集群接收方通道上指定此属性。 您在匹配的集群发送方通道上指定的任何均衡都可能被忽略。 请参阅 集群通道。

NPMSPEED (非持久消息速度)

此属性指定发送非持久消息的速度。

可能的值为:

NORMAL
通道上的非持久消息在事务中传输。

FAST
通道上的非持久消息不会在事务中传输。

缺省值为 FAST。 这样做的好处是可以更快地检索非持久消息。 缺点是，由于它们不是事务的一部分，因此如果发生传输故障，或者如果在传输消息时通道停止，那么可能会丢失消息。 请参阅 消息安全 (Safety of messages)。

注意:

1. z/OS 如果 IBM MQ for z/OS 的活动恢复日志切换和归档的频率高于预期，那么当通过通道发送的消息是非持久消息时，在通道的发送端和接收端设置 NPMSPEED (FAST) 可使 SYSTEM.CHANNEL.SYNCQ 更新。
2. 如果您看到与 SYSTEM.CHANNEL.SYNCQ，设置 NPMSPEED (FAST) 可以显着降低 CPU 使用率。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 接收方
- 请求者
- 集群发送方
- 集群接收方

密码（Password）

此属性指定在尝试启动与远程 MCA 的安全 LU 6.2 会话时， MCA 可以使用的密码。

您可以指定最大长度为 12 个字符的密码，尽管仅使用前 10 个字符。

z/OS 在 IBM MQ for z/OS 上，此属性仅对客户机连接通道有效。

Multi 在其他平台上，此属性对以下通道类型有效:

- 发送方
- 服务器
- 请求者
- 客户机连接
- 集群发送方

PORT (端口号)

指定用于连接 AMQP 客户机的端口号。

AMQP 1.0 连接的缺省端口是 5672。 如果您已使用端口 5672，那么可以指定其他端口。

PUTAUT (PUT 权限)

此属性指定要由 MCA 执行的安全性处理的类型。

使用此属性来选择要在执行时由 MCA 执行的安全性处理类型:

- 针对目标队列的 MQPUT 命令 (针对消息通道) ，或者
- MQI 调用 (针对 MQI 通道)。

z/OS 在 z/OS 上，检查的用户标识以及检查的用户标识数取决于 MQADMIN RACF 类 hlq.RESLEVEL 概要文件的设置。 根据通道启动程序的用户标识对 hlq.RESLEVEL 的访问级别，将检查一个或两个用户标识。 要查看检查的用户标识数，请参阅 RESLEVEL 和通道启动程序连接。 有关检查哪些用户标识的更多信息，请参阅 通道启动程序使用的用户标识。

您可以选择以下选项之一:

进程安全性，也称为缺省权限 (DEF)
将使用缺省用户标识。

Multi 在 Multiplatforms 版上，用于检查队列上打开权限的用户标识是在消息通道接收端运行 MCA 的进程或用户的用户标识。

z/OS 在 z/OS 上，可能会使用从网络接收的用户标识和从 MCAUSER 派生的用户标识，具体取决于要检查的用户标识数。

将使用此用户标识和打开选项 MQOO_SET_ALL_CONTEXT 打开队列。

上下文安全性 (CTX)
与消息关联的上下文信息中的用户标识将用作备用用户标识。

消息描述符中的 *UserIdentifier* 将移至对象描述符中的 *AlternateUserId* 字段。 使用打开选项 MQOO_SET_ALL_CONTEXT 和 MQOO_ALTERNATE_USER_AUTHORITY 打开队列。

Multi 在 Multiplatforms 版上，用于检查队列上 MQOO_SET_ALL_CONTEXT 和 MQOO_ALTERNATE_USER_AUTHORITY 的打开权限的用户标识是在消息通道接收端运行 MCA 的进程或用户的用户标识。 用于检查队列上 MQOO_OUTPUT 的打开权限的用户标识是消息描述符中的 *UserIdentifier* 。

z/OS 在 z/OS 上，可以使用从网络接收的用户标识或从 MCAUSER 派生的用户标识以及消息描述符中的上下文信息中的用户标识，具体取决于要检查的用户标识数。

在服务器连接通道上不支持上下文安全性 (CTX)。

z/OS **仅消息通道代理程序安全性 (ONLYMCA)**
将使用派生自 MCAUSER 的用户标识。

使用打开选项 MQOO_SET_ALL_CONTEXT 打开队列。

此值仅适用于 z/OS。

z/OS **备用消息通道代理程序安全性 (ALTMCA)**
可以使用消息描述符中的上下文信息 (*UserIdentifier* 字段) 中的用户标识以及派生自 MCAUSER 的用户标识，具体取决于要检查的用户标识数。

此值仅适用于 z/OS。

可以在 控制上下文信息中找到有关上下文字段和打开选项的更多详细信息。

可在此处找到有关安全性的更多信息:

- 保护
- ALW 在 AIX, Linux, and Windows 上设置安全性
- IBM i 在 IBM i 上设置安全性
- z/OS 在 z/OS 上设置安全性

此属性对以下通道类型有效:

- 接收方
- 请求者
- z/OS 服务器连接 (仅限 z/OS)
- 集群接收方

QMNAME (队列管理器名称)

其属性指定 IBM MQ MQI client 应用程序可请求连接的队列管理器或队列管理器组的名称。

此属性对以下通道类型有效:

- 客户机连接

QSGDISP (处置)

z/OS

此属性指定队列共享组中通道的处置。 它仅在 z/OS 上有效。

这些值是:

QMGR
在执行该命令的队列管理器的页集上定义通道。 此值是缺省值。

GROUP
此通道是在共享存储库中定义的。 仅当存在共享队列管理器环境时，才允许此值。 当使用 QSGDISP (GROUP) 定义通道时，将自动生成命令 DEFINE CHANNEL (name) NOREPLACE QSGDISP (COPY) 并将其发送到所有活动队列管理器，以使它们在页集 0 上生成本地副本。 对于未处于活动状态的队列管理器，或者以后加入队列共享组的队列管理器，将在队列管理器启动时生成该命令。

COPY
在执行该命令的队列管理器的页集上定义通道，并从同名的 QSGDISP (GROUP) 通道复制其定义。 仅当存在共享队列管理器环境时，才允许此值。

此属性对所有通道类型均有效。

RCVDATA (接收出口用户数据)

此属性指定传递到接收出口的用户数据。

您可以运行一系列接收出口。 一系列出口的用户数据字符串必须以逗号和/或空格分隔。 例如:

```
RCVDATA(exit1_data exit2_data)
MSGDATA(exit1_data,exit2_data)
SENDDATA(exit1_data, exit2_data)
```

ALW 在 IBM MQ for UNIX 系统和 Windows 系统中，出口名称字符串和用户数据字符串的长度限制为 500 个字符。

IBM i 在 IBM MQ for IBM i 中，最多可以指定 10 个出口名称，每个出口名称的用户数据长度限制为 32 个字符。

z/OS 在 IBM MQ for z/OS 中，最多可以指定 8 个用户数据字符串，每个字符串长度为 32 个字符。

此属性对所有通道类型均有效。

RCVEXIT (接收出口名称)

此属性指定要由通道接收用户出口运行的用户出口程序的名称。

此属性可以是要连续运行的程序的名称列表。 如果没有通道接收用户出口生效，请将其留空。

此属性的格式和最大长度取决于平台:

- z/OS 在 z/OS 上，它是最大长度为 8 个字符的装入模块名称，但最大长度为 128 个字符的客户机连接通道除外。
- IBM i 在 IBM i 上，格式如下:

  ```
  libname/progname
  ```

  在 CL 命令中指定时。

  当在 IBM MQ 命令 (MQSC) 中指定时，它具有以下格式:

  ```
  progname libname
  ```

  其中， *progname* 占用前 10 个字符， *libname* 占用后 10 个字符 (如果需要，请将两个空白填充到右侧)。 字符串的最大长度为 20 个字符。

- Linux AIX 在 AIX and Linux 上，格式如下:

```
libraryname(functionname)
```

字符串的最大长度为 40 个字符。

- Windows 在 Windows 上，格式如下:

```
dllname(functionname)
```

其中指定了 *dllname* ，但没有后缀 .DLL。 字符串的最大长度为 40 个字符。

z/OS 在 z/OS 上的集群发送方通道自动定义期间，通道出口名称将转换为 z/OS 格式。 如果要控制出口名称的转换方式，那么可以编写通道自动定义出口。 有关更多信息，请参阅 通道自动定义出口程序。

您可以指定接收，发送或消息出口程序名的列表。 这些名称必须以逗号和/或空格分隔。 例如:

```
RCVEXIT(exit1 exit2)
MSGEXIT(exit1,exit2)
SENDEXIT(exit1, exit2)
```

特定出口类型的出口名称字符串和用户数据字符串的总长度限制为 500 个字符。

- IBM i 在 IBM MQ for IBM i 上，最多可以列出 10 个出口名称。
- z/OS 在 IBM MQ for z/OS 上，最多可以列出八个出口名称。

此属性对所有通道类型均有效。

MQSC 关键字的通道属性 (S)

MQSC 关键字的通道属性的字母顺序列表，以字母 *S* 开头。

SCYDATA (安全出口用户数据)

此属性指定传递到安全出口的用户数据。

最大长度为 32 个字符。

此属性对所有通道类型均有效。

SCYEXIT (安全出口名称)

此属性指定要由通道安全出口运行的出口程序的名称。

如果没有有效的通道安全出口，就将其保留为空白。

名称的格式和最大长度取决于平台，如 RCVEXIT。 但是，只能指定一个安全出口。

有关安全出口的更多信息，请参阅 通道安全出口程序。

此属性对所有通道类型均有效。

SENDDATA (发送出口用户数据)

此属性指定传递到发送出口的用户数据。

您可以运行一系列发送出口。 有关用户数据长度的限制以及如何为多个出口指定 SENDDATA 的示例，如 RCVDATA 所示。 请参阅 RCVDATA。

此属性对所有通道类型均有效。

SENDEXIT (发送出口名称)

此属性指定要由通道发送出口运行的出口程序的名称。

此属性可以是要按顺序运行的程序的名称列表。 如果没有通道发送出口生效，请保留为空。

此属性的格式和最大长度取决于平台，如 RCVEXIT。

此属性对所有通道类型均有效。

SEQWRAP (序号换行)

此属性指定消息序号在 1 重新启动之前达到的最大数目。

该数字的值必须足够高，以避免在先前消息仍在使用该数字时重新发出该数字。 通道启动时，通道的两端必须具有相同的序号合并值; 否则，将发生错误。

该值可以设置为从 100 到 999 999 999。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 接收方
- 请求者
- 集群发送方
- 集群接收方

SHORTRTY (短重试计数)

此属性指定通道尝试将会话分配给其合作伙伴的最大次数。

可以将 SHORTRTY 属性从 0 设置为 999 999 999。

如果在通道中定义了多个 IP 地址，并且需要重新连接，那么 IBM MQ 将评估通道定义，并按定义顺序尝试连接到每个 IP 地址，直到成功建立连接或尝试所有地址为止。

在此情况下， SHORTRTY 与总体通道尝试重新连接的总尝试次数相关，而不是与单个 IP 地址相关

如果初始分配尝试失败，那么将减少 短重试计数 ，并且通道将在每次尝试之间以 **short retry interval** 属性中定义的时间间隔重试剩余次数。 如果它仍然失败，那么它将重试 *long retry count* 次，每次尝试之间的时间间隔为 *long retry interval* 。 如果仍未成功，那么通道将关闭。

z/OS 在 z/OS 上，如果已超过最大通道数 (**MAXCHL**) ，那么通道无法进入重试。

Multi 在 多平台上，为了尝试重试，通道启动程序必须正在运行。 通道启动程序必须监视在通道正在使用的传输队列定义中指定的启动队列。

如果 通道启动程序 (在 z/OS 上) 或者在通道正在重试时停止 通道 (在 多平台上) ，当通道启动程序或通道重新启动时，或者当在发送方通道成功放入消息时，将重置 短重试计数 和 长重试计数 。 但是，如果 通道启动程序 (在 z/OS 上) 或者 队列管理器 (在 Multiplatforms 版上) 关闭并重新启动，不会重置 短重试计数 和 长重试计数 。 通道保留在队列管理器重新启动或放入消息之前的重试计数值。

Multi 在 多平台 上:

1. 当通道从 RETRY 状态进入 RUNNING 状态时，不会立即重置 短重试计数 和 长重试计数 。 仅当通道进入 RUNNING 状态 (即本地通道确认发送到另一端的消息数) 后，第一条消息成功流经通道时，才会重置这些消息。
2. 重新启动通道时，将重置 短重试计数 和 长重试计数 。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 集群发送方
- 集群接收方

SHORTTMR (短重试时间间隔)

此属性指定在短重试方式下，通道在重试建立连接之前要等待的大致时间间隔 (以秒计)。

如果通道必须等待变为活动状态，那么重试之间的时间间隔可能会延长。

此属性可以设置为从零到 999 999 999。

如果在通道中定义了多个 IP 地址，并且需要重新连接，那么 IBM MQ 将评估通道定义，并按定义顺序尝试连接到每个 IP 地址，直到成功建立连接或尝试所有地址为止。

在这种情况下， SHORTTMR 与整体通道等待重新启动连接过程的时间有关，而不是与个别 IP 地址有关。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 集群发送方
- 集群接收方

SPLPROT (安全策略保护)

z/OS

此属性指定当 AMS 处于活动状态并且存在适用的策略时，服务器到服务器消息通道代理程序应如何处理消息保护。

此属性可设置为:

PASSTHRU
在发送方，服务器，接收方和请求者通道上

移除
在发送方和服务器通道上

ASPOLICY
在接收方和请求者通道上

此属性对以下通道类型有效:

- 发送方
- 服务器
- 接收方
- 请求者

SSLCAUTH (SSL 客户机认证)

SSLCAUTH 属性指定通道是否需要从 TLS 客户机接收 TLS 证书并对其进行认证。

SSLCAUTH 是可选属性。 此属性的可能值为:

可选
如果同级 TLS 客户机发送证书，那么将正常处理该证书，但如果未发送任何证书，那么认证不会失败。

必需
如果 TLS 客户机未发送证书，那么认证将失败。

缺省值为 REQUIRED。

可以在非 TLS 通道定义上指定 **SSLCAUTH** 的值。 即，缺少 SSLCIPH 属性的通道定义或该属性为空。

有关 SSLCAUTH 的更多信息，请参阅 DEFINE CHANNEL (MQTT) 和 保护。

SSLCAUTH 属性在所有可以接收通道启动流的通道类型上都有效，但发送方通道除外。 此属性对以下通道类型有效:

- 服务器

- 接收方
- 请求者
- 服务器连接
- 集群接收方

SSLCIPH (SSL 密码规范)

SSLCIPH 属性为 TLS 连接指定别名或单个名为 CipherSpec 的别名。

每个 IBM MQ 通道定义都包含 **SSLCIPH** 属性。 该值是最大长度为 32 个字符的字符串。

SSLCIPH 属性仅对传输类型 (**TRPTYPE**) 为 TCP 的通道有效。 如果 **TRPTYPE** 不是 TCP，那么将忽略数据，并且不会发出错误消息。

注意:

- **SSLCIPH** 属性可以包含空白值，这意味着您未使用 TLS。 如果通道的一端具有空白 **SSLCIPH** 属性，那么通道的另一端也必须具有空白 SSLCIPH 属性。

  Multi 如果启用了 SecureCommsOnly，则不支持纯文本通信，通道无法启动。
- 或者，如果 **SSLCIPH** 包含非空白值，那么该值可以是别名或指定的 CipherSpec。 信道协商信道两端都支持的最强兼容 CipherSpecs。
- 完全受管的 .NET 客户机可以指定特殊值 ***协商**。 此选项允许通道选择 .NET 框架支持的最新协议版本，并协商服务器支持的 CipherSpec 。

SSLCIPH 属性仅对传输类型 (**TRPTYPE**) 为 TCP 的通道有效。 如果 **TRPTYPE** 不是 TCP，那么将忽略数据，并且不会发出错误消息。

有关 **SSLCIPH** 的更多信息，请参阅 DEFINE CHANNEL 和 指定 CipherSpecs。

SSLPEER (SSL 对等项)

SSLPEER 属性用于在 IBM MQ 通道的另一端检查来自对等队列管理器或客户机的证书的专有名称 (DN)。

注: 通过与 TLS 主题专有名称进行匹配来限制通道连接的另一种方法是使用通道认证记录。 通过通道认证记录，可以将不同的 TLS 主题专有名称模式应用于同一通道。 如果通道上的 **SSLPEER** 和通道认证记录都用于应用于同一通道，那么入站证书必须与这两种模式匹配才能进行连接。

如果从同级接收到的 DN 与 **SSLPEER** 值不匹配，那么通道不会启动。

SSLPEER 是可选属性。 如果未指定值，那么启动通道时不会检查对等 DN。

SSLPEER 属性的最大长度取决于平台:

- z/OS 在 z/OS 上，该属性的最大长度为 256 个字节。
- Multi 在所有其他平台上，为 1024 字节。

通道认证记录在使用 **SSLPEER** 时提供了更大的灵活性，并且在所有平台上支持最大长度为 1024 个字节。

SSLPEER 属性值的检查还取决于平台:

- z/OS 在 z/OS 上，不会检查所使用的属性值。 如果输入不正确的值，那么通道会在启动时失败，并且会将错误消息写入通道两端的错误日志。 通道两端也会生成 "通道 SSL 错误" 事件。
- Multi 在支持 **SSLPEER** 的 多平台 上，首次输入字符串时将检查该字符串的有效性。

可以在非 TLS 通道定义上指定 **SSLPEER** 的值，在该定义上缺少 SSLCIPH 属性或该属性为空。 您可以使用此命令来临时禁用 TLS 以进行调试，而不必清除和稍后重新输入 TLS 参数。

SSLPEER 属性对所有通道类型都有效。

有关使用 **SSLPEER** 的更多信息，请参阅 SET CHLAUTH， 保护和 通道认证记录。

STATCHL (通道统计信息)

此属性控制通道的统计数据收集。

可能的值为:

QMGR
此通道的统计信息数据收集基于队列管理器属性 STATCHL 的设置。 该值为缺省值。

OFF
已禁用此通道的统计信息数据收集。

低
此通道的统计信息数据收集以较低的数据收集比率启用。

中型
此通道的统计信息数据收集已启用，数据收集比率适中。

高
此通道的统计信息数据收集是通过高比率的数据收集来启用的。

有关通道统计信息的更多信息，请参阅 监测参考。

z/OS 在 z/OS 系统上，无论您选择的值是多少，启用该参数都会简单地打开统计数据收集功能。 指定 LOW、 MEDIUM 或 HIGH 对结果没有影响。 必须启用此参数以收集通道记帐记录。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 接收方
- 请求者
- 集群发送方
- 集群接收方

MQSC 关键字的通道属性 (T-Z)

MQSC 关键字的通道属性的字母列表，以字母 *T* 到 *Z* 开头。

TPNAME (LU 6.2 事务程序名)

此属性用于 LU 6.2 连接。 它是要在链路远端运行的事务程序 (MCA) 的名称或通用名称。

当将辅助信息用于 SNA 通信时，事务程序名在 CPI-C 通信辅助对象或 APPC 辅助信息中定义，并且此属性必须留空。 否则，发送方通道和请求者通道需要此名称。

名称长度最多为 64 个字符。

必须将该名称设置为 SNA 事务程序名，除非 CONNAME 包含侧对象名，在这种情况下必须将其设置为空白。 实际名称取自 CPI-C 通信端对象或 APPC 端信息数据集。

此信息在不同平台上以不同方式设置; 请参阅 配置分布式排队 ，以获取有关为平台设置通信的更多信息。

此属性对以下通道类型有效:

- 发送方
- 服务器
- 请求者
- 客户机连接
- 集群发送方
- 集群接收方

TPROOT (主题根)

此属性指定 AMQP 通道的主题根。

您可以使用 TPROOT 属性来指定 AMQP 通道的主题根。 使用此属性可确保 MQ Light 应用程序在部署到队列管理器时，不会向其他应用程序正在使用的主题树区域发布消息或从这些区域预订消息。

TPROOT 的缺省值为 SYSTEM.BASE.TOPIC。 对于该值，AMQP 客户机用于发布或订阅的主题字符串没有前缀，且该客户机可以与其他 MQ 发布/订阅应用程序交换消息。 要使 AMQP 客户机在主题前缀下发布和预订，请先创建主题字符串设置为所需前缀的 MQ 主题对象，然后将 AMQP 通道 TPROOT 属性的值更改为您创建的 MQ 主题对象的名称。 以下示例显示了设置为 APPGROUP1.BASE.TOPIC :

```
DEFINE CHANNEL(MYAMQP) CHLTYPE(AMQP) TPROOT(APPGROUP1.BASE.TOPIC) PORT(5673)
```

注: 如果 TPROOT 属性值或支持它的主题字符串已更改，那么现有 AMQP 主题及其消息可能是孤立的。

TRPTYPE (传输类型)

此属性指定要使用的传输类型。

可能的值为:

表 *58:* 传输类型选项

值	**传输类型**
LU62	LU 6.2
TCP	TCP/IP
NETBIOS	NetBIOS 第 110 页的『1』
SPX	SPX 第 110 页的『1』

注意:

1. 用于 Windows。 还可以在 z/OS 上用于定义要在 Windows 上使用的客户机连接通道。

此属性对所有通道类型都有效，但被响应消息通道代理程序忽略。

USECLTID (使用客户机标识)

指定是否将客户机标识用于 AMQP 通道上的连接。 设置为 Yes 或 No。

USEDLQ (使用死信队列)

此属性确定当通道无法传递消息时是否使用死信队列 (或未传递的消息队列)。

可能的值为:

否
通道无法传递的消息将被视为失败。 根据 NPMSPEED 的设置，通道要么废弃这些消息，要么通道结束。

是（缺省值）
如果队列管理器 DEADQ 属性提供了死信队列的名称，那么将使用该队列，否则行为与 NO 相同。

USERID (用户标识)

此属性指定尝试启动与远程 MCA 的安全 SNA 会话时 MCA 要使用的用户标识。

您可以指定 20 个字符的任务用户标识。

在接收端，如果密码保持为加密格式，并且 LU 6.2 软件正在使用不同的加密方法，那么尝试启动通道将失败，但安全性详细信息无效。 可以通过将接收 SNA 配置修改为:

- 关闭密码替换，或者
- 定义安全用户标识和密码。

z/OS 在 IBM MQ for z/OS 上，此属性仅对客户机连接通道有效。

Multi 在其他平台上，此属性对以下通道类型有效:

- 发送方
- 服务器
- 请求者
- 客户机连接
- 集群发送方

XMITQ (传输队列名称)

此属性指定从中检索消息的传输队列的名称。

提供要与此发送方或服务器通道关联的传输队列的名称，该名称对应于该通道远侧的队列管理器。 您可以为传输队列提供与远程端的队列管理器相同的名称。

此属性对于类型为发送方或服务器的通道是必需的，对于其他通道类型无效。

IBM MQ 集群命令和属性

可以使用 MQSC 和 PCF 集群命令来刷新或重置集群，或者显示，恢复或暂挂集群队列管理器。 此外，用于定义通道，队列和队列管理器的 MQSC 和 PCF 命令具有适用于集群的属性。 其中一些属性由集群工作负载管理算法使用。

MQSC 命令

显示 MQSC 命令的方式与系统管理员在命令控制台上输入的方式相同。 请记住，您不必以这种方式发出命令。 根据您的平台，还有许多其他方法; 例如:

- IBM i 在 IBM MQ for IBM i 上，从 **WRKMQM** 的选项 26 以交互方式运行 MQSC 命令。 您还可以使用 CL 命令，或者可以将 MQSC 命令存储在文件中并使用 **STRMQMMQSC** CL 命令。
- z/OS 在 z/OS 上，可以使用 **CSQUTIL** 实用程序，操作和控制面板或 z/OS 控制台的 COMMAND 功能。
- 在所有其他平台上，可以将命令存储在文件中并使用 **runmqsc**。

在 MQSC 命令中，使用 CLUSTER 属性指定的集群名称可长达 48 个字符。

使用 CLUSNL 属性指定的集群名称列表最多可包含 256 个名称。 要创建集群名称列表，请使用 DEFINE NAMELIST 命令。

IBM MQ Explorer

IBM MQ Explorer GUI 可以在 IBM WebSphere MQ for z/OS 6.0 或更高版本上使用存储库队列管理器来管理集群。 您无需在单独的系统上指定其他存储库。 对于较早版本的 IBM MQ for z/OS， IBM MQ Explorer 无法使用存储库队列管理器来管理集群。 因此，您必须在 IBM MQ Explorer 可以管理的系统上指定其他存储库。

在 IBM MQ for Windows 和 IBM MQ for Linux 上，您还可以使用 IBM MQ Explorer 来处理集群。 您还可以使用独立 IBM MQ Explorer 客户机。

通过使用 IBM MQ Explorer，您可以查看集群队列并查询集群发送方和集群接收方通道的状态。 IBM MQ Explorer 包含两个向导，可用于指导您完成以下任务:

- 创建集群
- 将独立队列管理器连接到集群

专门用于处理集群的 MQSC 命令的 PCF 等效项

MQSC 命令	等效 PCF 命令
DISPLAY CLUSQMGR	MQCMD_INQUIRE_CLUSTER_Q_MGR
REFRESH CLUSTER	MQCMD_REFRESH_CLUSTER
RESET CLUSTER	MQCMD_RESET_CLUSTER
RESUME QMGR	MQCMD_RESUME_Q_MGR_CLUSTER
SUSPEND QMGR	MQCMD_SUSPEND_Q_MGR_CLUSTER

相关信息

集群：使用 REFRESH CLUSTER 最佳实践

通道定义命令上可用的集群属性

可在通道定义命令上指定的集群属性。

DEFINE CHANNEL，ALTER CHANNEL 和 DISPLAY CHANNEL 命令具有针对集群的两个特定 CHLTYPE 参数: CLUSRCVR 和 CLUSSDR。 要定义集群接收方通道，请使用 DEFINE CHANNEL 命令并指定 CHLTYPE(CLUSRCVR)。 集群接收方通道定义上的许多属性与接收方或发送方通道定义上的属性相同。 要定义集群发送方通道，请使用 DEFINE CHANNEL 命令，指定 CHLTYPE(CLUSSDR)以及用于定义发送方通道的许多相同属性。

在定义集群发送方通道时，不再需要指定完整存储库队列管理器的名称。 如果您知道用于集群中通道的命名约定，那么可以使用 +QMNAME+ 构造来生成 CLUSSDR 定义。 z/OS 上不支持 +QMNAME+ 构造。 连接后，IBM MQ 会更改通道的名称，并替换正确的完整存储库队列管理器名称以代替 +QMNAME+。 生成的通道名称将截断为 20 个字符。

有关命名约定的更多信息，请参阅 集群命名约定。

仅当命名通道的约定包含队列管理器的名称时，该技术才有效。 例如，使用名为 CLUSTER1.QM1.ALPHA 的集群接收方通道在名为 CLUSTER1 的集群中定义名为 QM1 的完整存储库队列管理器。 其他每个队列管理器都可以使用通道名称 CLUSTER1.+QMNAME+.ALPHA 来定义此队列管理器的集群发送方通道。

如果对所有通道使用相同的命名约定，请注意一次只能存在一个 +QMNAME+ 定义。

DEFINE CHANNEL 和 ALTER CHANNEL 命令上的以下属性特定于集群通道:

cluster
CLUSTER 属性指定与此通道关联的集群的名称。 或者，使用 CLUSNL 属性。

CLUSNL
CLUSNL 属性指定集群名称的名称列表。

NETPRTY
仅集群接收方。

NETPRTY 属性指定通道的网络优先级。 NETPRTY 可帮助工作负载管理例程。 如果有多个到目标的可能路径，那么工作负载管理例程将选择具有最高优先级的路径。

CLWLPRTY
出于工作负载管理目的， CLWLPRTY 参数将优先级因子应用于同一目标的通道。 此参数指定用于集群工作负载分发的通道优先级。 该值必须在范围 0 到 9 中，其中 0 是最低优先级， 9 是最高优先级。

CLWLRANK
CLWLRANK 参数将排名因子应用于通道以进行工作负载管理。 此参数指定用于集群工作负载分布的通道的列组。 该值必须在范围 0 到 9 中，其中 0 是最低等级， 9 是最高等级。

CLWLWGHT
CLWLWGHT 参数将加权因子应用于通道以进行工作负载管理。 CLWLWGHT 对通道进行加权，以便可以控制该通道下发送的消息所占的比例。 集群工作负载算法使用 CLWLWGHT 来偏差目标选项，以便可以通过特定通道发送更多消息。 缺省情况下，所有通道权重属性都是相同的缺省值。 weight 属性允许您在功

能强大的 UNIX 机器上分配比小型桌面 PC 上的其他通道更大的权重。 权重越大，表示集群工作负载算法比 PC 更频繁地选择 UNIX 机器作为消息的目标。

CONNAME
集群接收方通道定义上指定的 CONNAME 在整个集群中用于标识队列管理器的网络地址。 请注意为 CONNAME 参数选择一个在整个 IBM MQ 集群中解析的值。 请勿使用通用名称。 请记住，在集群接收方通道上指定的值优先于在相应集群发送方通道中指定的任何值。

DEFINE CHANNEL 命令和 ALTER CHANNEL 命令上的这些属性也适用于 DISPLAY CHANNEL 命令。

注: 自动定义的集群发送方通道从接收队列管理器上的相应集群接收方通道定义中获取其属性。 即使存在手动定义的集群发送方通道，也会自动修改其属性，以确保它们与相应集群接收方定义上的属性匹配。 例如，请注意，您可以在不指定 CONNAME 参数中的端口号的情况下定义 CLUSRCVR，同时手动定义指定端口号的 CLUSSDR。 当自动定义的 CLUSSDR 替换手动定义的端口号时，端口号 (取自 CLUSRCVR) 变为空白。 将使用缺省端口号，并且通道将失败。

注: DISPLAY CHANNEL 命令不会显示自动定义的通道。 但是，您可以使用 DISPLAY CLUSQMGR 命令来检查自动定义的集群发送方通道的属性。

使用 DISPLAY CHSTATUS 命令可显示集群发送方或集群接收方通道的状态。 此命令提供手动定义的通道和自动定义的通道的状态。

等效的 PCF 为 MQCMD_CHANGE_CHANNEL， MQCMD_COPY_CHANNEL， MQCMD_CREATE_CHANNEL 和 MQCMD_INQUIRE_CHANNEL。

在 CLUSRCVR 定义上省略 CONNAME 值

在某些情况下，可以省略 CLUSRCVR 定义上的 CONNAME 值。 不得在 z/OS 上省略 CONNAME 值。

Multi 在 多平台 上，集群接收方通道的 TCP/IP 连接名称参数是可选的。 如果您将连接名称留空，IBM MQ 将为您生成一个连接名称，采用缺省端口并使用系统的当前 IP 地址。 您可以覆盖缺省端口号，但仍使用系统的当前 IP 地址。 对于每个连接名称，请将 IP 名称留空，并在括号内提供端口号；例如：

```
(1415)
```

生成的 **CONNAME** 始终采用点分十进制 (IPv4) 或十六进制 (IPv6) 格式，而不采用字母数字 DNS 主机名形式。

当您有使用动态主机配置协议 (DHCP) 的机器时，此工具很有用。 如果在 CLUSRCVR 通道上未提供 CONNAME 的值，那么不需要更改 CLUSRCVR 定义。 DHCP 为您分配新的 IP 地址。

如果在 CLUSRCVR 定义上为 CONNAME 指定空白，那么 IBM MQ 将从系统的 IP 地址生成 CONNAME。 仅生成的 CONNAME 存储在存储库中。 集群中的其他队列管理器不知道 CONNAME 最初为空。

如果发出 DISPLAY CLUSQMGR 命令，那么您将看到生成的 CONNAME。 但是，如果从本地队列管理器发出 DISPLAY CHANNEL 命令，那么会看到 CONNAME 为空。

如果队列管理器由于 DHCP 而停止并使用其他 IP 地址重新启动，那么 IBM MQ 将重新生成 CONNAME 并相应地更新存储库。

相关概念
集群中的工作负载均衡
如果集群包含同一队列的多个实例，那么 IBM MQ 会选择要将消息路由到的队列管理器。 它使用集群工作负载管理算法和许多特定于集群工作负载的属性来确定要使用的最佳队列管理器。

z/OS Asynchronous behavior of CLUSTER commands on z/OS
The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

相关参考
队列定义命令上可用的集群属性
可以在队列定义命令上指定的集群属性。

队列管理器定义命令上可用的集群属性
可以在队列管理器定义命令上指定的集群属性。

DISPLAY CLUSQMGR
使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

刷新集群
从队列管理器发出 REFRESH CLUSTER 命令以废弃有关集群的所有本地保存的信息。 除非在特殊情况下，否则您不太可能需要使用此命令。

重置集群: 强制从集群中移除队列管理器
在特殊情况下，使用 **RESET CLUSTER** 命令从集群中强制除去队列管理器。

SUSPEND QMGR， RESUME QMGR 和集群
使用 SUSPEND QMGR 和 RESUME QMGR 命令来临时减少此队列管理器的入站集群活动 (例如，在此队列管理器上执行维护之前)，然后将其恢复。

第 123 页的『集群工作负载均衡-通道属性』
集群工作负载均衡中使用的通道属性的字母顺序列表。

队列定义命令上可用的集群属性

可以在队列定义命令上指定的集群属性。

DEFINE QLOCAL，DEFINE QREMOTE 和 DEFINE QALIAS 命令

DEFINE QLOCAL， DEFINE QREMOTE 和 DEFINE QALIAS 命令以及三个等效 ALTER 命令上的集群属性为:

cluster
指定队列所属的集群的名称。

CLUSNL
指定集群名称的名称列表。

DefBind
指定当应用程序在 MQOPEN 调用上指定 MQOO_BIND_AS_Q_DEF 时要使用的绑定。 此属性的选项为:

- 指定 DEFBIND(OPEN) 以在打开队列时将队列句柄绑定到集群队列的特定实例。 DEFBIND(OPEN) 是此属性的缺省值。
- 指定 DEFBIND(NOTFIXED)，以便队列句柄不与集群队列的任何实例绑定。
- 指定 DEFBIND(GROUP) 以允许应用程序请求将一组消息全部分配给同一目标实例。

在队列管理器集群中公布多个同名队列时，应用程序可以选择是否将此应用程序中的所有消息发送到单个实例 (MQOO_BIND_ON_OPEN)，以允许工作负载管理算法根据每条消息选择最合适的目标 (MQOO_BIND_NOT_FIXED)，或者允许应用程序请求将消息的 "组" 全部分配给同一目标实例 (MQOO_BIND_ON_GROUP)。 在消息组之间重新驱动工作负载均衡 (无需队列的 MQCLOSE 和 MQOPEN)。

在队列定义上指定 DEFBIND 时，将使用属性 MQBND_BIND_ON_OPEN， MQBND_BIND_NOT_FIXED 或 MQBND_BIND_ON_GROUP 之一来定义队列。 使用群组时，必须指定 MQBND_BIND_ON_OPEN 或 MQBND_BIND_ON_GROUP 。

在同一集群队列的所有实例上，应该将 DEFBIND 属性设置为相同的值。

CLWLRANK
将排名因子应用于队列以进行工作负载管理。 CLWLRANK 参数在模型队列上不受支持。 集群工作负载算法选择具有最高列组的目标队列。 缺省情况下，所有队列的 CLWLRANK 设置为零。

如果最终目标是另一个集群上的队列管理器，那么可以在相邻集群的交集处设置任何中间网关队列管理器的列组。 通过对中间队列管理器进行排名，集群工作负载算法可以正确选择距离最终目标更近的目标队列管理器。

相同的逻辑适用于别名队列。 在检查通道状态之前进行列组选择，因此即使不可访问的队列管理器也可供选择。 这具有允许通过网络路由消息的效果，而不是让它在两个可能的目标之间进行选择 (就像优先级一样)。 因此，如果通道未启动到列组所指示的位置，那么消息不会路由到下一个最高列组，而是等到通道可用于该目标 (消息保留在传输队列上)。

CLWLPRTY
将优先级因子应用于队列以进行工作负载管理。 集群工作负载算法选择具有最高优先级的目标队列。 缺省情况下，所有队列的优先级都设置为零。

如果有两个可能的目标队列，那么可以使用此属性将一个目标故障转移到另一个目标。 在检查通道状态后进行优先级选择。 将所有消息发送到具有最高优先级的队列，除非到该目标的通道的状态不如到其他目标的通道的状态有利。 这意味着只有最容易访问的目标可供选择。 这将影响所有可用的多个目标之间的优先级划分。

CLWLUSEQ
指定队列的 MQPUT 操作的行为。 此参数指定当目标队列具有本地实例和至少一个远程集群实例 (MQPUT 源自集群通道的情况除外) 时 MQPUT 操作的行为。 此参数仅对本地队列有效。

可能的值包括: QMGR (行为由队列管理器定义的 CLWLUSEQ 参数指定) ， ANY (队列管理器将本地队列视为集群队列的另一个实例，用于工作负载分布) ， LOCAL (本地队列是 MQPUT 操作的唯一目标，前提是已启用本地队列)。 MQPUT 行为取决于 集群工作负载管理算法。

DISPLAY QUEUE 和 DISPLAY QCLUSTER 命令

DEFINE QLOCAL， DEFINE QREMOTE 和 DEFINE QALIAS 命令上的属性也适用于 DISPLAY QUEUE 命令。

要显示有关集群队列的信息，请在 DISPLAY QUEUE 命令上指定队列类型 QCLUSTER 或关键字 CLUSINFO ，或者使用命令 DISPLAY QCLUSTER。

DISPLAY QUEUE 或 DISPLAY QCLUSTER 命令返回主管队列的队列管理器的名称 (如果有多个队列实例，那么返回所有队列管理器的名称)。 它还返回主管队列的每个队列管理器的系统名称，表示的队列类型以及定义可供本地队列管理器使用的日期和时间。 使用 CLUSQMGR， QMID， CLUSQT， CLUSDATE 和 CLUSTIME 属性返回此信息。

队列管理器 (QMID) 的系统名称是系统为队列管理器生成的唯一名称。

您可以定义同样是共享队列的集群队列。 例如， 在 z/OS 上，您可以定义:

```
DEFINE QLOCAL(MYQUEUE) CLUSTER(MYCLUSTER) QSGDISP(SHARED) CFSTRUCT(STRUCTURE)
```

等效的 PCF 为 MQCMD_CHANGE_Q， MQCMD_COPY_Q， MQCMD_CREATE_Q 和 MQCMD_INQUIRE_Q。

相关概念
集群中的工作负载均衡
如果集群包含同一队列的多个实例，那么 IBM MQ 会选择要将消息路由到的队列管理器。 它使用集群工作负载管理算法和许多特定于集群工作负载的属性来确定要使用的最佳队列管理器。

z/OS Asynchronous behavior of CLUSTER commands on z/OS
The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

相关参考
通道定义命令上可用的集群属性
可在通道定义命令上指定的集群属性。

队列管理器定义命令上可用的集群属性
可以在队列管理器定义命令上指定的集群属性。

DISPLAY CLUSQMGR
使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

刷新集群
从队列管理器发出 REFRESH CLUSTER 命令以废弃有关集群的所有本地保存的信息。 除非在特殊情况下，否则您不太可能需要使用此命令。

重置集群: 强制从集群中移除队列管理器
在特殊情况下，使用 **RESET CLUSTER** 命令从集群中强制除去队列管理器。

SUSPEND QMGR， RESUME QMGR 和集群

使用 SUSPEND QMGR 和 RESUME QMGR 命令来临时减少此队列管理器的入站集群活动 (例如，在此队列管理器上执行维护之前)，然后将其恢复。

第 124 页的『集群工作负载均衡-队列属性』
集群工作负载均衡中使用的队列属性的字母顺序列表。

队列管理器定义命令上可用的集群属性

可以在队列管理器定义命令上指定的集群属性。

要指定队列管理器保存集群的完整存储库，请使用指定属性 REPOS(*clustername*)的 **ALTER QMGR** 命令。 要指定多个集群名称的列表，请定义集群名称列表，然后在 **ALTER QMGR** 命令上指定属性 REPOSNL(*namelist*)：

```
DEFINE NAMELIST(CLUSTERLIST)
       DESCR('List of clusters whose repositories I host')
       NAMES(CLUS1, CLUS2, CLUS3)
 ALTER QMGR REPOSNL(CLUSTERLIST)
```

您可以在 ALTER QMGR 命令上提供其他集群属性

CLWLEXIT(*name*)
指定将消息放入集群队列时要调用的用户出口的名称。

CLWLDATA(*data*)
指定要传递到集群工作负载用户出口的数据。

CLWLLEN(*length*)
指定要传递到集群工作负载用户出口的最大消息数据量

CLWLMRUC(*channels*)
指定出站集群通道的最大数量。

CLWLMRUC 是未在集群中传播的本地队列管理器属性。 它可用于集群工作负载出口和选择消息目标的集群工作负载算法。

CLWLUSEQ(LOCAL|ANY)
指定当目标队列同时具有本地实例和至少一个远程集群实例时 MQPUT 的行为。 如果放置操作起源于集群通道，那么此属性不适用。 可以将 CLWLUSEQ 同时指定为队列属性和队列管理器属性。

如果指定 ANY，那么本地队列和远程队列都是 MQPUT 的可能目标。

如果指定 LOCAL，那么本地队列是 MQPUT 的唯一目标。

等效的 PCF 为 MQCMD_CHANGE_Q_MGR 和 MQCMD_INQUIRE_Q_MGR。

相关概念

集群中的工作负载均衡
如果集群包含同一队列的多个实例，那么 IBM MQ 会选择要将消息路由到的队列管理器。 它使用集群工作负载管理算法和许多特定于集群工作负载的属性来确定要使用的最佳队列管理器。

z/OS Asynchronous behavior of CLUSTER commands on z/OS
The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

相关参考

通道定义命令上可用的集群属性
可在通道定义命令上指定的集群属性。

队列定义命令上可用的集群属性
可以在队列定义命令上指定的集群属性。

DISPLAY CLUSQMGR
使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

刷新集群
从队列管理器发出 REFRESH CLUSTER 命令以废弃有关集群的所有本地保存的信息。 除非在特殊情况下，否则您不太可能需要使用此命令。

重置集群: 强制从集群中移除队列管理器
在特殊情况下，使用 **RESET CLUSTER** 命令从集群中强制除去队列管理器。

SUSPEND QMGR， RESUME QMGR 和集群
使用 SUSPEND QMGR 和 RESUME QMGR 命令来临时减少此队列管理器的入站集群活动 (例如，在此队列管理器上执行维护之前)，然后将其恢复。

第 125 页的『集群工作负载均衡-队列管理器属性』
集群工作负载均衡中使用的队列管理器属性的字母顺序列表。

DISPLAY CLUSQMGR

使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

如果从具有完整存储库的队列管理器发出此命令，那么返回的信息适用于集群中的每个队列管理器。 否则，返回的信息仅适用于其关注的队列管理器。 即，其尝试将消息发送至的每个队列管理器，以及持有完整存储库的每个队列管理器。

此信息包含适用于集群发送方和集群接收方通道的大多数通道属性。 此外，还可以显示以下属性:

CHANNEL
队列管理器的集群接收方通道名称。

CLUSDATE
定义可供本地队列管理器使用的日期。

cluster
队列管理器的集群。

CLUSTIME
定义可供本地队列管理器使用的时间。

DEFTYPE
队列管理器的定义方式。 DEFTYPE 可以是下列其中一个值:

CLUSSDR
已在本地队列管理器上以管理方式定义集群发送方通道，但目标队列管理器尚未识别该集群发送方通道。 要处于此状态，本地队列管理器已定义手动集群发送方通道，但接收队列管理器未接受集群信息。 这可能是由于由于可用性或集群发送方配置中的错误 (例如，发送方与接收方定义之间的 CLUSTER 属性不匹配) 导致从未建立通道。 这是暂时性的情况或错误状态，应该进行调查。

CLUSSDRA
此值表示自动发现的集群队列管理器，未在本地定义集群发送方通道。 这是集群队列管理器的 DEFTYPE，对于这些集群队列管理器，本地队列管理器没有本地配置，但已通知这些集群队列管理器。 例如

- 如果本地队列管理器是完整存储库队列管理器，那么应该是集群中所有部分存储库队列管理器的 DEFTYPE 值。
- 如果本地队列管理器是部分存储库，那么这可能是正在从此本地队列管理器使用的集群队列的主机，也可能是已通知此队列管理器使用的第二个完整存储库队列管理器的主机。

如果 DEFTYPE 值为 CLUSSDRA，并且本地和远程队列管理器都是指定集群的完整存储库，那么配置不正确，因为必须定义本地定义的集群发送方通道才能将其转换为 CLUSSDRB 的 DEFTYPE。

CLUSSDRB
集群发送方通道已在本地队列管理器上以管理方式定义，并且已被目标队列管理器接受为有效集群通道。 这是部分存储库队列管理器的手动配置的完整存储库队列管理器的预期 DEFTYPE。 它还应该是集群中从一个完整存储库到另一个完整存储库的任何 CLUSQMGR 的 DEFTYPE。 不应将手动集群发送方通道配置为部分存储库或从部分存储库队列管理器配置为多个完整存储库。 如果在上述任一情况下看到 CLUSSDRB 的 DEFTYPE，那么应该对其进行调查和更正。

CLUSRCVR
以管理方式定义为本地队列管理器上的集群接收方通道。 这表示集群中的本地队列管理器。

注: 要确定哪些 CLUSQMGRs 是集群的完整存储库队列管理器，请参阅 QMTYPE 属性。

有关定义集群通道的更多信息，请参阅 集群通道。

QMTYPE
是保存完整存储库还是仅保存部分存储库。

状态
此队列管理器的集群发送方通道的状态。

SUSPEND
是否暂挂队列管理器。

版本
集群队列管理器所关联的 IBM MQ 安装的版本。

版本的格式为 VVRRMMFF：

- VV：版本
- RR：发行版
- MM：维护级别
- FF：修订级别

XMITQ
队列管理器所使用的集群传输队列。

另请参阅 DISPLAY QCLUSTER 命令。 这在 DISPLAY QUEUE 以及 第 114 页的『队列定义命令上可用的集群属性』的 DISPLAY QUEUE 和 DISPLAY QCLUSTER 命令 部分中简要描述。 有关使用 DISPLAY QCLUSTER 的示例，请搜索 “显示 Qcluster” 和 “DIS QCLUSTER”的信息集。

相关概念

集群中的工作负载均衡
如果集群包含同一队列的多个实例，那么 IBM MQ 会选择要将消息路由到的队列管理器。 它使用集群工作负载管理算法和许多特定于集群工作负载的属性来确定要使用的最佳队列管理器。

z/OS Asynchronous behavior of CLUSTER commands on z/OS
The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

相关参考

通道定义命令上可用的集群属性
可在通道定义命令上指定的集群属性。

队列定义命令上可用的集群属性
可以在队列定义命令上指定的集群属性。

队列管理器定义命令上可用的集群属性
可以在队列管理器定义命令上指定的集群属性。

刷新集群
从队列管理器发出 REFRESH CLUSTER 命令以废弃有关集群的所有本地保存的信息。 除非在特殊情况下，否则您不太可能需要使用此命令。

重置集群: 强制从集群中移除队列管理器
在特殊情况下，使用 **RESET CLUSTER** 命令从集群中强制除去队列管理器。

SUSPEND QMGR， RESUME QMGR 和集群
使用 SUSPEND QMGR 和 RESUME QMGR 命令来临时减少此队列管理器的入站集群活动 (例如，在此队列管理器上执行维护之前) ，然后将其恢复。

MQSC 命令 **DISPLAY CLUSQMGR**

刷新集群

从队列管理器发出 REFRESH CLUSTER 命令以废弃有关集群的所有本地保存的信息。 除非在特殊情况下，否则您不太可能需要使用此命令。

此命令有三种形式:

REFRESH CLUSTER(clustername) REPOS(NO)
缺省值。 队列管理器将保留所有本地定义的集群队列管理器和集群队列以及作为完整存储库的所有集群队列管理器的知识。 此外，如果队列管理器是集群的完整存储库，那么它还会保留对集群中其他集群队列管理器的了解。 其他所有内容都将从存储库的本地副本中除去，并从集群中的其他完整存储库中重新构建。 如果使用 REPOS(NO)，那么不会停止集群通道。 完整存储库使用其 CLUSSDR 通道来通知集群的其余部分已完成刷新。

REFRESH CLUSTER(clustername) REPOS(YES)
除了缺省行为外，还会刷新表示完整存储库集群队列管理器的对象。 如果队列管理器是完整存储库，那么使用此选项无效，如果使用此命令，那么将失败并记录错误 AMQ9406/CSQX406E 。 如果它是完整存储库，那么必须首先对其进行变更，以使其不是有关集群的完整存储库。 将从手动定义的 CLUSSDR 定义中恢复完整存储库位置。 发出使用 REPOS(YES) 刷新后，可以更改队列管理器，使其再次成为完整存储库 (如果需要)。

REFRESH CLUSTER(*)
刷新其所属的所有集群中的队列管理器。 如果与 REPOS(YES) REFRESH CLUSTER(*) 配合使用，那么会产生强制队列管理器从本地 CLUSSDR 定义中的信息重新启动其搜索完整存储库的额外效果。 即使 CLUSSDR 通道将队列管理器连接到多个集群，也会执行搜索。

注: 对于大型集群，使用 **REFRESH CLUSTER** 命令可能会对正在运行的集群造成干扰，此外每隔 27 天在集群对象向所有相关队列管理器自动发送状态更新时也可能有干扰。 请参阅在大型集群中刷新可能会影响集群的性能和可用性。

相关概念

集群中的工作负载均衡
如果集群包含同一队列的多个实例，那么 IBM MQ 会选择要将消息路由到的队列管理器。 它使用集群工作负载管理算法和许多特定于集群工作负载的属性来确定要使用的最佳队列管理器。

z/OS Asynchronous behavior of CLUSTER commands on z/OS
The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

相关参考

通道定义命令上可用的集群属性
可在通道定义命令上指定的集群属性。

队列定义命令上可用的集群属性
可以在队列定义命令上指定的集群属性。

队列管理器定义命令上可用的集群属性
可以在队列管理器定义命令上指定的集群属性。

DISPLAY CLUSQMGR
使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

重置集群: 强制从集群中移除队列管理器
在特殊情况下，使用 **RESET CLUSTER** 命令从集群中强制除去队列管理器。

SUSPEND QMGR, RESUME QMGR 和集群
使用 SUSPEND QMGR 和 RESUME QMGR 命令来临时减少此队列管理器的入站集群活动 (例如，在此队列管理器上执行维护之前)，然后将其恢复。

相关信息

集群：使用 REFRESH CLUSTER 最佳实践

重置集群: 强制从集群中移除队列管理器

在特殊情况下，使用 **RESET CLUSTER** 命令从集群中强制除去队列管理器。

除非在特殊情况下，否则您不太可能需要使用此命令。

只能从完整存储库队列管理器发出 **RESET CLUSTER** 命令。 此命令采用两种格式，具体取决于您是按名称还是按标识引用队列管理器。

1. `RESET CLUSTER(clustername) QMNAME(qmname) ACTION(FORCEREMOVE) QUEUES(NO)`

2. `RESET CLUSTER(clustername) QMID(qmid) ACTION(FORCEREMOVE) QUEUES(NO)`

不能同时指定 QMNAME 和 QMID。 如果使用 QMNAME，并且集群中有多个具有该名称的队列管理器，那么不会运行该命令。 使用 QMID 而不是 QMNAME 来确保运行 **RESET CLUSTER** 命令。

在 **RESET CLUSTER** 命令上指定 QUEUES(NO) 是缺省值。 指定 QUEUES(YES) 将从集群中除去对队列管理器拥有的集群队列的引用。 除了从集群本身中除去队列管理器外，还会除去这些引用。

即使集群队列管理器在集群中不可见，也会除去这些引用; 原因可能是先前在没有 QUEUE 选项的情况下强制除去了这些引用。

例如，如果已删除队列管理器，但仍对集群定义了集群接收方通道，那么可以使用 **RESET CLUSTER** 命令。 您可以发出 **RESET CLUSTER** 命令以更快地进行整理，而不是等待 IBM MQ 除去这些定义 (它会自动除去这些定义)。 然后，将通知集群中的所有其他队列管理器该队列管理器不再可用。

如果队列管理器暂时损坏，那么您可能希望在集群中的其他队列管理器尝试向其发送消息之前告知这些队列管理器。 **RESET CLUSTER** 将除去损坏的队列管理器。 稍后，当受损队列管理器再次工作时，请使用 **REFRESH CLUSTER** 命令来反转 **RESET CLUSTER** 的效果，并将队列管理器返回到集群。如果队列管理器位于发布/预订集群中，那么您需要恢复任何必需的代理预订。 请参阅针对发布/预订集群的 REFRESH CLUSTER 注意事项。

注: 对于大型集群，使用 **REFRESH CLUSTER** 命令可能会对正在运行的集群造成干扰，此外每隔 27 天在集群对象向所有相关队列管理器自动发送状态更新时也可能有干扰。 请参阅在大型集群中刷新可能会影响集群的性能和可用性。

使用 **RESET CLUSTER** 命令是删除自动定义的集群发送方通道的唯一方法。

要点: 如果要除去的自动定义的通道处于不确定状态，那么 **RESET CLUSTER** 不会立即除去该通道。 在此情况下，您需要在 **RESET CLUSTER** 命令之前发出 RESOLVE CHANNEL 命令。

在正常情况下，您不太可能需要此命令。 IBM 支持人员可能会建议您发出该命令，以整理集群队列管理器所保存的集群信息。 请勿将此命令用作从集群中除去队列管理器的简短命令。 从集群中除去队列管理器中描述了从集群中除去队列管理器的正确方法。

由于存储库仅保留信息 90 天，因此在该时间之后，被强制除去的队列管理器可以重新连接到集群。 它会自动重新连接，除非已将其删除。 如果要阻止队列管理器重新加入集群，那么需要采取相应的安全措施。

所有集群命令 (**DISPLAY CLUSQMGR** 除外) 都以异步方式工作。 用于更改涉及集群的对象属性的命令将更新对象并向存储库处理器发送请求。 检查用于处理集群的命令的语法，并将请求发送到存储库处理器。

将异步处理发送到存储库处理器的请求以及从集群的其他成员接收的集群请求。 如果必须将它们传播到整个集群以确定它们是否成功，那么处理可能需要相当长的时间。

相关概念

集群中的工作负载均衡
如果集群包含同一队列的多个实例，那么 IBM MQ 会选择要将消息路由到的队列管理器。 它使用集群工作负载管理算法和许多特定于集群工作负载的属性来确定要使用的最佳队列管理器。

z/OS Asynchronous behavior of CLUSTER commands on z/OS
The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

相关参考

通道定义命令上可用的集群属性
可在通道定义命令上指定的集群属性。

队列定义命令上可用的集群属性
可以在队列定义命令上指定的集群属性。

队列管理器定义命令上可用的集群属性
可以在队列管理器定义命令上指定的集群属性。

DISPLAY CLUSQMGR
使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

刷新集群

从队列管理器发出 REFRESH CLUSTER 命令以废弃有关集群的所有本地保存的信息。 除非在特殊情况下，否则您不太可能需要使用此命令。

SUSPEND QMGR， RESUME QMGR 和集群
使用 SUSPEND QMGR 和 RESUME QMGR 命令来临时减少此队列管理器的入站集群活动 (例如，在此队列管理器上执行维护之前)，然后将其恢复。

RESET CLUSTER (重置集群)

SUSPEND QMGR， RESUME QMGR 和集群

使用 SUSPEND QMGR 和 RESUME QMGR 命令来临时减少此队列管理器的入站集群活动 (例如，在此队列管理器上执行维护之前)，然后将其恢复。

当队列管理器从集群中暂挂时，如果集群中的备用队列管理器上存在同名的可用队列，那么它不会在其托管的集群队列上接收消息。 但是，明确以此队列管理器为目标的消息，或者目标队列仅在此队列管理器上可用的消息，仍将定向到此队列管理器。

通过停止此集群的集群接收方通道，可以防止在队列管理器暂挂时接收更多入站消息。 要停止集群的集群接收方通道，请使用 SUSPEND QMGR 命令的 FORCE 方式。

相关概念
集群中的工作负载均衡
如果集群包含同一队列的多个实例，那么 IBM MQ 会选择要将消息路由到的队列管理器。 它使用集群工作负载管理算法和许多特定于集群工作负载的属性来确定要使用的最佳队列管理器。

z/OS Asynchronous behavior of CLUSTER commands on z/OS
The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

相关任务
维护队列管理器
相关参考
通道定义命令上可用的集群属性
可在通道定义命令上指定的集群属性。

队列定义命令上可用的集群属性
可以在队列定义命令上指定的集群属性。

队列管理器定义命令上可用的集群属性
可以在队列管理器定义命令上指定的集群属性。

DISPLAY CLUSQMGR
使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

刷新集群
从队列管理器发出 REFRESH CLUSTER 命令以废弃有关集群的所有本地保存的信息。 除非在特殊情况下，否则您不太可能需要使用此命令。

重置集群: 强制从集群中移除队列管理器
在特殊情况下，使用 **RESET CLUSTER** 命令从集群中强制除去队列管理器。

已暂挂的队列管理器
恢复队列管理器

集群中的工作负载均衡

如果集群包含同一队列的多个实例，那么 IBM MQ 会选择要将消息路由到的队列管理器。 它使用集群工作负载管理算法和许多特定于集群工作负载的属性来确定要使用的最佳队列管理器。

集群工作负载管理算法根据队列管理器和队列的可用性以及与通道，队列和队列管理器相关联的许多特定于集群工作负载的属性来选择合适的目标。 这些属性在子主题中进行了描述。

在配置特定于集群工作负载的属性之后，如果配置行为与预期不同，请探索算法如何选择队列管理器的详细信息。 请参阅第 126 页的『集群工作负载管理算法』。 如果此算法的结果不满足您的需求，那么您可以编

写集群工作负载用户出口程序，并使用此出口将消息路由到集群中您选择的队列。 请参阅 编写和编译集群工作负载出口。

表 59: 特定于集群工作负载的属性的摘要	
属性名称	**描述**
通道属性	
CLWLPRTY	指定集群工作负载分布的通道的优先级顺序。
CLWLRANK	指定集群工作负载分布的通道的排名。
CLWLWGHT	指定应用于集群工作负载分布的 CLUSSDR 和 CLUSRCVR 通道的权重。
NETPRTY	指定 CLUSRCVR 通道的优先级。
队列属性	
CLWLPRTY	指定集群工作负载分布的本地队列，远程队列或别名队列的优先级。
CLWLRANK	指定用于集群工作负载分发的本地队列，远程队列或别名队列的列组。
CLWLUSEQ	指定是否优先于集群中的其他队列实例，使用队列本地实例作为目标。
队列管理器属性	
CLWLMRUC	设置最近选择的通道数。
CLWLUSEQ	指定是否将队列的本地实例作为目标优先于集群中队列的其他实例。

相关概念

z/OS Asynchronous behavior of CLUSTER commands on z/OS
The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

相关参考

通道定义命令上可用的集群属性
可在通道定义命令上指定的集群属性。

队列定义命令上可用的集群属性
可以在队列定义命令上指定的集群属性。

队列管理器定义命令上可用的集群属性
可以在队列管理器定义命令上指定的集群属性。

DISPLAY CLUSQMGR
使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

刷新集群
从队列管理器发出 REFRESH CLUSTER 命令以废弃有关集群的所有本地保存的信息。 除非在特殊情况下，否则您不太可能需要使用此命令。

重置集群: 强制从集群中移除队列管理器
在特殊情况下，使用 **RESET CLUSTER** 命令从集群中强制除去队列管理器。

SUSPEND QMGR， RESUME QMGR 和集群
使用 SUSPEND QMGR 和 RESUME QMGR 命令来临时减少此队列管理器的入站集群活动 (例如，在此队列管理器上执行维护之前)，然后将其恢复。

集群工作负载均衡-通道属性

集群工作负载均衡中使用的通道属性的字母顺序列表。

注: 在目标队列管理器上的集群接收方通道上指定集群工作负载通道属性。 您在匹配的集群发送方通道上指定的任何均衡都可能被忽略。 请参阅 集群通道 (Cluster channel)。

CLWLPRTY (集群工作负载优先级)

CLWLPRTY 通道属性指定了群集工作负载分配通道的优先级顺序。 该值必须在 0-9 范围内，其中 0 是最低优先级， 9 是最高优先级。

使用 "**CLWLPRTY** 通道属性为可用的群集目的地设置优先顺序。 IBM MQ 先选择优先级最高的目的地，然后再选择集群目的地优先级最低的目的地。 如果存在多个具有相同优先级的目标，那么它将选择最近最少使用的目标。

如果有两个可能的目标，那么可以使用此属性来允许故障转移。 消息转至具有最高优先级通道的队列管理器。 如果它变为不可用，那么消息将转至下一个优先级最高的队列管理器。 优先级较低的队列管理器充当保留。

IBM MQ 先检查通道状态，然后再确定通道的优先级。 只有可用的队列管理器才是可供选择的候选者。

注意:

- 在目标队列管理器上的集群接收方通道上指定此属性。 您在匹配的集群发送方通道上指定的任何均衡都可能被忽略。 请参阅 集群通道。
- 远程队列管理器的可用性取决于该队列管理器的通道状态。 当通道启动时，它们的状态会发生多次更改，其中一些状态对集群工作负载管理算法的优先级较低。 在实践中，这意味着可以在启动高优先级 (主) 目标的通道时选择低优先级 (备份) 目标。
- 如果需要确保没有信息发送到备份目的地，请不要使用 "**CLWLPRTY**。 考虑使用独立队列或 "**CLWLRANK**，从主队列手动切换到备用队列。

CLWLRANK (集群工作负载列组)

CLWLRANK 通道属性指定集群工作负载分布的通道的排名。 该值必须在 0-9 范围内，其中 0 是最低等级， 9 是最高等级。

如果要控制发送到另一个集群中的队列管理器的消息的最终目标，请使用 **CLWLRANK** 通道属性。 通过设置将队列管理器连接到集群交集处的网关队列管理器的通道的列组来控制最终目标的选择。

设置 **CLWLRANK** 时，消息将通过相互连接的集群采用指定的路由，以到达排名较高的目标。 例如，消息到达网关队列管理器，该网关队列管理器可以使用排名 1 和 2 的通道将它们发送到两个队列管理器中的任何一个。 它们将自动发送到由具有最高列组的通道连接的队列管理器，在这种情况下，会将该通道发送到排名 2 的队列管理器。

IBM MQ 在检查通道状态之前获取通道的排名。 在检查通道状态之前获取列组意味着即使不可访问的通道也可供选择。 它允许通过网络路由消息，即使最终目标不可用也是如此。

注意:

- 在目标队列管理器上的集群接收方通道上指定此属性。 您在匹配的集群发送方通道上指定的任何均衡都可能被忽略。 请参阅 集群通道。
- 如果还使用了优先级属性 **CLWLPRTY**，那么 IBM MQ 将在可用目标之间进行选择。 如果通道不可用于具有最高列组的目标，那么消息将保留在传输队列上。 当通道变为可用时，将释放此消息。 消息不会按列组顺序发送到下一个可用目标。

CLWLWGHT (集群工作负载权重)

CLWLWGHT 通道属性指定应用于 CLUSSDR 和 CLUSRCVR 通道的权重，用于群集工作负载分配。 该值必须在 1-99 范围内，其中 1 是最低权重， 99 是最高权重。

使用 "**CLWLWGHT** 可向处理能力更强的服务器发送更多信息。 通道权重越高，通过该通道发送的消息越多。

注意:

- 在目标队列管理器上的集群接收方通道上指定此属性。 您在匹配的集群发送方通道上指定的任何均衡都可能被忽略。 请参阅 集群通道。
- 当 "**CLWLWGHT** 从任何通道的默认值 50 变更时，工作量平衡将取决于发送到任何群集队列的报文选择每个通道的总次数。 有关更多信息，请参阅 第 126 页的『集群工作负载管理算法』。

NETPRTY (网络连接优先级)

NETPRTY 通道属性指定 CLUSRCVR 通道的优先级。 该值必须在 0-9 范围内，其中 0 是最低优先级， 9 是最高优先级。

使用 NETPRTY 属性使一个网络成为主网络，另一个网络成为备份网络。 给定一组排名相同的通道，集群会在多个路径可用时选择具有最高优先级的路径。

使用 NETPRTY 通道属性的典型示例是区分具有不同成本或速度并连接相同目标的网络。

注: 在目标队列管理器上的集群接收方通道上指定此属性。 您在匹配的集群发送方通道上指定的任何均衡都可能被忽略。 请参阅 集群通道。

相关概念

集群工作负载管理算法
工作负载管理算法使用工作负载均衡属性和许多规则来选择要放入集群队列的消息的最终目标。

相关参考

集群工作负载均衡-队列属性
集群工作负载均衡中使用的队列属性的字母顺序列表。

集群工作负载均衡-队列管理器属性
集群工作负载均衡中使用的队列管理器属性的字母顺序列表。

第 112 页的『通道定义命令上可用的集群属性』
可在通道定义命令上指定的集群属性。

集群工作负载均衡-队列属性

集群工作负载均衡中使用的队列属性的字母顺序列表。

CLWLPRTY

CLWLPRTY 队列属性指定集群工作负载分发的本地队列，远程队列或别名队列的优先级。 该值必须在 0-9 范围内，其中 0 是最低优先级， 9 是最高优先级。

使用 **CLWLPRTY** 队列属性来设置目标队列的首选项。 IBM MQ 先选择具有最高优先级的目标，然后再选择具有最低集群目标优先级的目标。 如果存在多个具有相同优先级的目标，那么它将选择最近最少使用的目标。

IBM MQ 在检查通道状态后获取队列管理器的优先级。 只有可用的队列管理器才是可供选择的候选者。

注:

远程队列管理器的可用性取决于该队列管理器的通道状态。 当通道启动时，它们的状态会发生多次更改，其中一些状态对集群工作负载管理算法的优先级较低。 在实践中，这意味着可以在启动高优先级 (主) 目标的通道时选择低优先级 (备份) 目标。

如果需要确保没有消息转至备份目标，请勿使用 **CLWLPRTY**。 请考虑使用单独的队列，或者使用 **CLWLRANK** 从主队列到备份的手动切换。

如果有两个可能的目标，那么可以使用此属性来允许故障转移。 优先级最高的队列管理器接收请求，优先级较低的队列管理器充当保留。 如果最高优先级队列管理器失败，那么可用的下一个最高优先级队列管理器将接管。

CLWLRANK

CLWLRANK 队列属性指定集群工作负载分布的本地队列，远程队列或别名队列的列组。 该值必须在 0-9 范围内，其中 0 是最低等级， 9 是最高等级。

如果要控制发送到另一个集群中的队列管理器的消息的最终目标，请使用 **CLWLRANK** 队列属性。 设置 **CLWLRANK** 时，消息将通过相互连接的集群采用指定的路由，以到达排名较高的目标。

例如，您可能定义了两个配置完全相同的网关队列管理器，以提高网关的可用性。 假设您已在网关上为集群中定义的本地队列定义了集群别名队列。 如果本地队列变得不可用，那么您打算将消息保留在其中一个网关上，以等待队列再次变为可用。 要保留网关上的队列，必须定义具有高于网关上的集群别名队列的列组的本地队列。

如果定义与队列别名具有相同列组的本地队列，并且本地队列不可用，那么消息将在网关之间传输。 当发现本地队列不可用时，第一个网关队列管理器会将消息路由到另一个网关。 另一个网关尝试再次将消息传递到目标本地队列。 如果本地队列仍然不可用，那么它会将消息路由回第一个网关。 消息将在网关之间来回移动，直到目标本地队列再次可用为止。 通过为本地队列提供更高的列组，即使队列不可用，也不会将消息重新路由到较低列组的目标。

IBM MQ 在检查通道状态之前获取队列的列组。 在检查通道状态之前获取列组意味着即使不可访问的队列也可供选择。 它允许通过网络路由消息，即使最终目标不可用也是如此。

如果使用了优先级属性 IBM MQ，请在可用目标之间进行选择。 如果通道不可用于具有最高列组的目标，那么消息将保留在传输队列上。 当通道变为可用时，将释放此消息。 消息不会按列组顺序发送到下一个可用目标。

CLWLUSEQ

CLWLUSEQ 队列属性指定是否将队列的本地实例作为目标优先于集群中的其他实例。

CLWLUSEQ 队列属性仅对本地队列有效。 仅当消息由应用程序或不是集群通道的通道放入时才适用。

Local
本地队列是 MQPUT 的唯一目标，前提是已启用本地队列。 MQPUT 行为取决于 集群工作负载管理。

QMGR
行为由 **CLWLUSEQ** 队列管理器属性指定。

ANY
MQPUT 将本地队列视为与集群中用于工作负载分发的任何其他队列实例相同。

相关概念

集群工作负载管理算法
工作负载管理算法使用工作负载均衡属性和许多规则来选择要放入集群队列的消息的最终目标。

相关参考

集群工作负载均衡-通道属性
集群工作负载均衡中使用的通道属性的字母顺序列表。

集群工作负载均衡-队列管理器属性
集群工作负载均衡中使用的队列管理器属性的字母顺序列表。

第 114 页的『队列定义命令上可用的集群属性』
可以在队列定义命令上指定的集群属性。

集群工作负载均衡-队列管理器属性

集群工作负载均衡中使用的队列管理器属性的字母顺序列表。

CLWLMRUC

CLWLMRUC 队列管理器属性设置最近选择的通道数。 集群工作负载管理算法使用 **CLWLMRUC** 来限制活动出站集群通道数。 该值必须在范围 1-999 999 999 之间。

初始缺省值为 999 999 999 999。

CLWLUSEQ

CLWLUSEQ 队列管理器属性指定是否将队列的本地实例作为目标优先于集群中队列的其他实例。 如果 **CLWLUSEQ** 队列属性设置为 QMGR，那么此属性适用。

CLWLUSEQ 队列属性仅对本地队列有效。 仅当消息由应用程序或不是集群通道的通道放入时才适用。

Local
本地队列是 MQPUT 的唯一目标。 LOCAL 是缺省值。

ANY
MQPUT 将本地队列视为与集群中用于工作负载分发的任何其他队列实例相同。

相关概念
集群工作负载管理算法
工作负载管理算法使用工作负载均衡属性和许多规则来选择要放入集群队列的消息的最终目标。

相关参考
集群工作负载均衡-通道属性
集群工作负载均衡中使用的通道属性的字母顺序列表。

集群工作负载均衡-队列属性
集群工作负载均衡中使用的队列属性的字母顺序列表。

第 116 页的『队列管理器定义命令上可用的集群属性』
可以在队列管理器定义命令上指定的集群属性。

集群工作负载管理算法

工作负载管理算法使用工作负载均衡属性和许多规则来选择要放入集群队列的消息的最终目标。

每次需要选择目标时，都会执行工作负载管理算法:

- 它在使用 MQOO_BIND_ON_OPEN 选项打开集群队列时使用。
- 每次使用 MQOO_BIND_NOT_FIXED 打开消息时，都会将该消息放入集群队列。
- 当 MQOO_BIND_ON_GROUP 用于打开集群队列时，每次启动新消息组时都将使用此消息组。
- 对于 主题主机路由，每次将消息发布到集群主题时都会使用此消息。 如果本地队列管理器不是此主题的主机，那么将使用算法来选择要路由消息的主机队列管理器。

以下部分描述了在确定要放入集群队列的消息的最终目标时使用的工作负载管理算法。 这些规则受应用于队列，队列管理器和通道的以下属性的设置影响:

表 *60:* 集群工作负载管理的属性

队列	队列管理器	通道
• CLWLPRTY [1] • CLWLRANK [1] • CLWLUSEQ [1] • PUT / PUB	• CLWLMRUC • CLWLUSEQ [1]	• CLWLPRTY • CLWLRANK • CLWLWGHT • 网络属性

最初，队列管理器从两个过程构建可能的目标列表:

- 将目标 ObjectName 和 ObjectQmgrName 与队列管理器所在集群中共享的队列管理器别名定义进行匹配。
- 查找到队列管理器的唯一路由 (即通道) ， 该队列管理器托管名为 ObjectName 的队列，并且位于队列管理器所属的其中一个集群中。

算法将逐步执行以下规则，以从可能的目标列表中除去目标。

1. 将消除未与本地队列管理器共享集群的队列或主题或远程 CLUSRCVR 通道的远程实例。
2. 如果指定了队列或主题名称，那么将消除与队列或主题不在同一集群中的远程 CLUSRCVR 通道。

   注: 此阶段的所有剩余队列，主题和通道都可供集群工作负载出口使用 (如果已配置)。

[1] 此属性仅在选择集群队列时适用，而在选择主题时不适用。

3. 将消除 CLWLRANK 小于所有剩余通道或队列管理器别名的最大列组的所有队列管理器或队列管理器别名的通道。
4. 将消除 CLWLRANK 小于所有剩余队列的最大列组的所有队列 (不是队列管理器别名)。
5. 如果保留了队列，主题或队列管理器别名的多个实例，并且启用了 pub put ，那么将消除所有已禁用 put 的实例。

   注: 如果仅保留已禁用的实例，那么仅查询操作将成功，所有其他操作都将失败，并带有 MQRC_CLUSTER_PUT_DISABLED。
6. 选择队列时，如果生成的队列集包含队列的本地实例，那么通常使用本地实例。 如果满足下列其中一个条件，那么将使用队列的本地实例:
   - 队列 CLWLUSEQ 的 use-queue 属性设置为 LOCAL。
   - 以下两个语句均为 true:
     - 队列 CLWLUSEQ 的 use-queue 属性设置为 QMGR。
     - 队列管理器 CLWLUSEQ 的 use-queue 属性设置为 LOCAL。
   - 通过集群通道接收消息，而不是由本地应用程序放入消息。
   - 对于使用 CLWLUSEQ (ANY) 定义的本地定义队列，或者从队列管理器继承同一设置的本地定义队列，在适用的更广泛的条件集中，满足以下条件:
     - 根据队列所在集群中本地定义的 CLUSRCVR 通道的 `status` ，选择本地队列。 此状态将与 CLUSSDR 通道的状态进行比较，这些通道会将消息传递到远程定义的同名队列。

       例如，在与队列相同的集群中有一个 CLUSRCVR。 该 CLUSRCVR 具有 "正在停止" 状态，而集群中同名的其他队列具有 "正在运行" 或 "不活动" 状态。 在这种情况下，将选择远程通道，而不使用本地 CLUSSDR 通道。
     - 本地队列是根据 CLUSRCVR 通道的 `number` 选择的，在与具有相同状态的 CLUSSDR 通道的任何比较中，会将消息带到远程定义的同名队列。

       例如，在与队列相同的集群中有四个 CLUSRCVR 通道和一个 CLUSSDR 通道。 所有通道都具有相同的 INACTIVE 或 RUNNING 状态。 因此，有五个通道可供选择，还有两个队列实例。 五分之四 (80%) 的消息转至本地队列。
7. 如果保留了多个队列管理器，那么如果未暂挂任何队列管理器，那么将消除所有已暂挂的队列管理器。
8. 如果保留队列或主题的多个远程实例，那么将包括处于不活动状态或正在运行的所有通道。 列出了状态常量:
   - MQCHS_INACTIVE
   - MQCHS_RUNNING
9. 如果未保留队列或主题的远程实例，那么将包括处于绑定，初始化，启动或停止状态的所有通道。 列出了状态常量:
   - MQCHS_BINDING
   - MQCHS_初始化
   - 正在启动 MQCHS_STAR 丁
   - MQCHS_正在停止
10. 如果没有保留队列或主题的远程实例，那么将包括所有正在重试的通道。 列出了状态常量:
    - MQCHS_RE 试用
11. 如果没有保留队列或主题的远程实例，那么将包括处于 "正在请求" ， "已暂停" 或 "已停止" 状态的所有通道。 列出了状态常量:
    - MQCHS_请求
    - MQCHS_PAUSED
    - MQCHS_STOPPED
    - MQCHS_交换机

12. 如果任何队列管理器上的队列或主题的多个远程实例仍然存在，那么将选择每个队列管理器的 NETPRTY 值最高的通道。
13. 将除去除具有最高优先级 CLWLPRTY 的通道和别名以外的所有剩余通道和队列管理器别名。 如果保留任何队列管理器别名，那么将保留到队列管理器的通道。
14. 如果正在选择队列:
    - 将消除除具有最高优先级 CLWLPRTY 的队列以外的所有队列，并保留通道。
15. 然后，通过消除具有最低 MQWDR.DestSeqNumber。

    注: 内部集群控制消息在适当情况下使用相同的集群工作负载算法发送。

计算有效目标的列表后，将使用以下逻辑在消息之间进行工作负载均衡:

- 当一个目标的多个远程实例保留，并且到该目标的所有通道都将 CLWLWGHT 设置为缺省设置 50 时，将选择最近最少使用的通道。 这大约等同于存在多个远程实例时工作负载均衡的循环法样式。
- 当一个目标的多个远程实例仍然存在，并且这些队列的一个或多个通道将 CLWLWGHT 设置为非缺省设置 (即使它们都具有匹配的非缺省值) 时，那么路由将取决于每个通道的相对权重以及发送消息时先前选择每个通道的总次数。
- 观察具有多个实例的单个集群队列的消息分布时，这可能导致在队列实例的子集之间分布不平衡。 这是因为要平衡的是此队列管理器中每个集群发送方通道的历史使用，而不仅仅是该队列的消息流量。 如果不需要此行为，请完成下列其中一个步骤:
  - 如果需要均匀分布，请在所有集群接收方通道上将 CLWLWGHT 设置为 50。
  - 或者，如果某些队列实例需要与其他队列实例进行不同的加权，请使用定义的专用集群接收方通道在专用集群中定义这些队列。 此操作将这些队列的工作负载均衡与集群中的其他队列隔离。
- 如果更改了可用集群接收方通道的任何集群工作负载属性，或者集群接收方通道的状态变为可用，那么将重置用于均衡通道的历史数据。 修改手动定义的集群发送方通道的工作负载属性不会重置历史数据。
- 在考虑集群工作负载出口逻辑时，所选通道是具有最低 MQWDR.DestSeqFactor。 每次选择通道时，此值将增加大约 1000/CLWLWGHT。 如果存在多个具有最低值的通道，那么其中一个具有最低 MQWDR.DestSeqNumber 值。

用户消息的分发并非始终准确，因为集群的管理和维护会导致消息在通道之间流动。 结果是用户消息分布不均，可能需要一些时间才能稳定下来。 由于管理消息和用户消息混合在一起，因此在工作负载均衡期间不依赖消息的精确分布。

相关参考

集群工作负载均衡-通道属性
集群工作负载均衡中使用的通道属性的字母顺序列表。

集群工作负载均衡-队列属性
集群工作负载均衡中使用的队列属性的字母顺序列表。

集群工作负载均衡-队列管理器属性
集群工作负载均衡中使用的队列管理器属性的字母顺序列表。

Asynchronous behavior of CLUSTER commands on z/OS

The command issuer of a cluster command on z/OS receives confirmation a command has been sent, but not that it has completed successfully.

For both REFRESH CLUSTER and RESET CLUSTER, message CSQM130I is sent to the command issuer indicating that a request has been sent. This message is followed by message CSQ9022I to indicate that the command has completed successfully, in that a request has been sent. It does not indicate that the cluster request has been completed successfully.

Any errors are reported to the z/OS console on the system where the channel initiator is running, they are not sent to the command issuer.

The asynchronous behavior is in contrast to CHANNEL commands. A message indicating that a channel command has been accepted is issued immediately. At some later time, when the command has been completed, a message indicating either normal or abnormal completion is sent to the command issuer.

Related concepts

集群中的工作负载均衡

如果集群包含同一队列的多个实例，那么 IBM MQ 会选择要将消息路由到的队列管理器。 它使用集群工作负载管理算法和许多特定于集群工作负载的属性来确定要使用的最佳队列管理器。

Related tasks

Checking that async commands for distributed networks have finished

Related reference

通道定义命令上可用的集群属性

可在通道定义命令上指定的集群属性。

队列定义命令上可用的集群属性

可以在队列定义命令上指定的集群属性。

队列管理器定义命令上可用的集群属性

可以在队列管理器定义命令上指定的集群属性。

DISPLAY CLUSQMGR

使用 DISPLAY CLUSQMGR 命令可显示有关集群中队列管理器的集群信息。

刷新集群

从队列管理器发出 REFRESH CLUSTER 命令以废弃有关集群的所有本地保存的信息。 除非在特殊情况下，否则您不太可能需要使用此命令。

重置集群: 强制从集群中移除队列管理器

在特殊情况下，使用 **RESET CLUSTER** 命令从集群中强制除去队列管理器。

SUSPEND QMGR， RESUME QMGR 和集群

使用 SUSPEND QMGR 和 RESUME QMGR 命令来临时减少此队列管理器的入站集群活动 (例如，在此队列管理器上执行维护之前)，然后将其恢复。

通道程序

此部分查看可用于通道的不同类型的通道程序 (MCA)。

下表显示了 MCA 的名称。

表 *61: AIX, Linux, and Windows* 系统的通道程序

程序名	**连接方向**	**通信**
阿姆克姆帕		任意
运行 mqlsr	入站	任意
amqcrs6a	入站	LU 6.2
阿姆克斯塔	入站	TCP
鲁姆奇尔	出站	任意
鲁姆基	出站	任意

runmqlsr (运行 IBM MQ 侦听器)， runmqchl (运行 IBM MQ 通道) 和 runmqchi (运行 IBM MQ 通道启动程序) 是可以在命令行中输入的控制命令。

将使用 inetd 对 AIX and Linux 系统上的 TCP 通道调用 amqcrsta，其中不启动任何侦听器。

使用 LU6.2 时，会将 amqcrs6a 作为事务程序调用

IBM i IBM i 上的相互通信作业

以下作业与 IBM i 上的 Intercommunication 相关联。 这些名称包含在下表中。

表 62: 作业名和描述	
作业名	**描述**
AMQCLMAA	非线程化侦听器
AMQCRSTA	非线程化响应程序作业
AMQRMPPA	通道池作业
运行 MQCHI	通道启动程序
运行 MQCHL	通道作业
运行 MQLSR	线程化侦听器

IBM i IBM i 上的通道状态

通道状态显示在 "使用通道" 面板上

表 63: *IBM i* 上的通道状态	
州名	**含义**
STARTING	通道已准备好开始与目标 MCA 协商
BINDING	建立会话和初始数据交换
REQUESTING	发起连接的请求者通道
RUNNING	正在传输或准备传输
PAUSED	等待消息重试时间间隔
STOPPING	确定是重试还是停止
RETRYING	正在等待下次重试
STOPPED	由于错误或由于发出了结束通道命令而停止通道
不活动的	通道已正常结束处理或通道从未启动
* 无	无状态 (仅适用于服务器连接通道)

ALW 示例: 在 AIX, Linux, and Windows 上规划消息通道

此信息提供了一个详细示例，说明如何将两个队列管理器连接在一起，以便可以在它们之间发送消息。

关于此任务

在所有示例中， MQSC 命令显示为它们将出现在命令文件中，并且它们将在命令行上输入。 这两种方法看起来完全相同，但是，要在命令行上发出命令，必须首先为缺省队列管理器输入 `runmqsc`，或者输入 `runmqsc` *qmname* ，其中 *qmname* 是所需队列管理器的名称。 然后输入任意数量的命令，如示例中所示。

另一种方法是创建包含这些命令的文件。 然后，命令中的任何错误都很容易更正。 如果您调用了文件 mqsc.in ，那么要在队列管理器 QMNAME 上运行该文件，请使用:

```
runmqsc QMNAME < mqsc.in > mqsc.out
```

您可以先验证文件中的命令，然后再使用以下命令运行该文件:

```
runmqsc -v QMNAME < mqsc.in > mqsc.out
```

对于可移植性，应将命令的行长度限制为 72 个字符。 使用并置字符在多行上继续。 在 Windows 上，使用 Ctrl-z 在命令行上结束输入。 在 AIX and Linux 系统上，使用 Ctrl-d。 或者，使用 **end** 命令。

第 131 页的图 7 显示了示例方案。

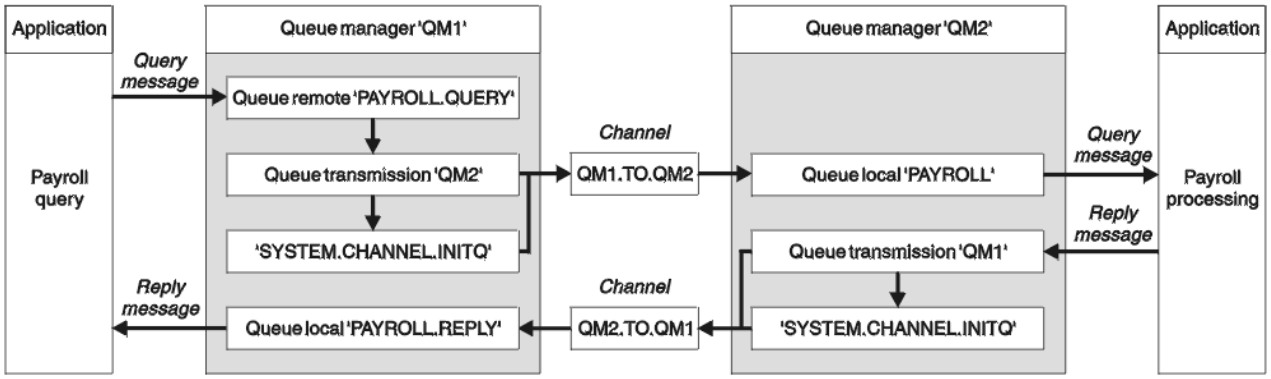

图 7: *AIX, Linux, and Windows* 系统的消息通道示例

此示例涉及连接到队列管理器 QM1 的薪资查询应用程序，该应用程序将薪资查询消息发送到在队列管理器 QM2 上运行的薪资处理应用程序。 工资单查询应用程序需要对其发送回 QM1 的查询的应答。 薪资查询消息在名为 QM1.TO.QM2 的发送方/接收方通道上从 QM1 发送到 QM2 ，应答消息在另一个名为 QM2.TO.QM1 的发送方/接收方通道上从 QM2 发送回 QM1 。 一旦这两个通道有消息要发送到另一个队列管理器，就会触发这两个通道启动。

工资单查询应用程序将查询消息放入 QM1 上定义的远程队列 “PAYROLL.QUERY” 。 此远程队列定义解析为 QM2 上的本地队列 “PAYROLL” 。 此外，工资单查询应用程序指定将对查询的应答发送到 QM1 上的本地队列 “PAYROLL.REPLY” 。 工资单处理应用程序从 QM2 上的本地队列 “工资单” 获取消息，并将应答发送到所需的任何位置; 在此情况下，本地队列 “PAYROLL.REPLY” 位于 QM1 上。

在 TCP/IP 的示例定义中， QM1 的主机地址为 192.0.2.0 ，并且正在侦听端口 1411 ，而 QM2 的主机地址为 192.0.2.1 ，并且正在侦听端口 1412。 此示例假定这些内容已在您的系统上定义并且可供使用。

需要在 QM1 上创建的对象定义包括:

- 远程队列定义， PAYROLL.QUERY
- 传输队列定义， QM2 (缺省值 = 远程队列管理器名称)
- 发送方通道定义， QM1.TO.QM2
- 接收方通道定义， QM2.TO.QM1
- 应答队列定义 PAYROLL.REPLY

需要在 QM2 上创建的对象定义包括:

- 本地队列定义， PAYROLL
- 传输队列定义， QM1 (缺省值 = 远程队列管理器名称)
- 发送方通道定义， QM2.TO.QM1
- 接收方通道定义， QM1.TO.QM2

连接详细信息在发送方通道定义的 CONNAME 属性中提供。

您可以在 第 131 页的图 7 中查看排列图。

过程

请参阅:

- 第 131 页的『设置 AIX, Linux, and Windows 的消息通道示例』 以获取有关设置消息通道的详细信息
- 第 133 页的『运行和展开 AIX, Linux, and Windows 的示例』 以获取有关如何使用其他产品 (例如 CICS) 以及如何连接更多应用程序和用户出口的建议。

ALW 设置 AIX, Linux, and Windows 的消息通道示例

这些对象定义允许连接到队列管理器 QM1 的应用程序将请求消息发送到 QM2 上名为 PAYROLL 的队列，并在 QM1 上名为 PAYROLL.REPLY 的队列上接收应答，并允许连接到队列管理器 QM2 的应用程序从名为

PAYROLL 的本地队列检索请求消息，并将这些请求消息的应答放入队列管理器 QM1 上名为 PAYROLL.REPLY 的队列。

关于此任务

已向所有对象定义提供 DESCR 和 REPLACE 属性。 提供的其他属性是使示例起作用所需的最低值。 未提供的属性采用队列管理器 QM1 和 QM2 的缺省值。

您不需要提供远程队列定义以允许将应答返回到 QM1。 从本地队列 PAYROLL 中检索的消息的消息描述符同时包含应答队列和应答队列管理器名称。 因此，只要 QM2 可以将应答队列管理器名称解析为队列管理器 QM2 上传输队列的名称，就可以发送应答消息。 在此示例中，应答队列管理器名称为 QM1 ，因此队列管理器 QM2 需要同名的传输队列。

过程

- 在队列管理器 QM1:
  a) 设置远程队列定义:

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QM2') REPLACE +
PUT(ENABLED) XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
```

  注: 远程队列定义不是物理队列，而是将消息定向到传输队列 QM2 的一种方法，以便可以将这些消息发送到队列管理器 QM2。

  b) 设置传输队列定义:

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(QM1.TO.QM2.PROCESS)
```

  将第一条消息放入此传输队列时，会将触发器消息发送到启动队列 SYSTEM.CHANNEL.INITQ。 通道启动程序从启动队列中获取消息，并启动在指定进程中标识的通道。

  c) 设置发送方通道定义:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +
CONNAME('192.0.2.1(1412)')
```

  d) 设置接收方通道定义:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QM2')
```

  e) 设置 reply-to_queue 定义:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to query messages sent to QM2')
```

  应答队列定义为 PUT (ENABLED)。 这将确保可以将应答消息放入队列。 如果无法将应答放入应答队列，那么会将其发送至 QM1 上的死信队列，如果此队列不可用，请将其保留在队列管理器 QM2 上的传输队列 QM1 上。 队列已定义为 GET (ENABLED) ，以允许检索应答消息。

- 在队列管理器 QM2:
  a) 设置本地队列定义:

```
DEFINE QLOCAL(PAYROLL) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Local queue for QM1 payroll details')
```

  由于与队列管理器 QM1 上的应答队列定义相同的原因，此队列定义为 PUT (ENABLED) 和 GET (ENABLED)。

  b) 设置传输队列定义:

```
DEFINE QLOCAL(QM1) DESCR('Transmission queue to QM1') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(QM2.TO.QM1.PROCESS)
```

将第一条消息放入此传输队列时，会将触发器消息发送到启动队列 SYSTEM.CHANNEL.INITQ。 通道启动程序从启动队列中获取消息，并启动在指定进程中标识的通道。

c) 设置发送方通道定义:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +
CONNAME('192.0.2.0(1411)')
```

d) 设置接收方通道定义:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QM1')
```

ALW 运行和展开 AIX, Linux, and Windows 的示例

有关启动通道启动程序和侦听器的信息以及有关在此场景中展开的建议。

关于此任务

一旦创建了这些定义，您就需要:

- 在每个队列管理器上启动通道启动程序。
- 启动每个队列管理器的侦听器。

您还可以展开该示例。

过程

1. 启动通道启动程序和侦听器。
   请参阅 为 Windows 设置通信和 在 AIX and Linux 系统上设置通信。
2. 您可以通过以下方法展开此示例:
   - 使用 LU 6.2 通信与 CICS 系统互连以及事务处理。
   - 添加更多队列，进程和通道定义，以允许其他应用程序在两个队列管理器之间发送消息。
   - 在通道上添加用户出口程序以允许进行链接加密，安全检查或其他消息处理。
   - 使用队列管理器别名和应答队列别名来了解有关如何在队列管理器网络的组织中使用这些别名的更多信息。

IBM i 示例: 在 IBM i 上规划消息通道

有关如何将两个 IBM i 队列管理器连接在一起以便可以在它们之间发送消息的详细示例。 此示例涉及连接到队列管理器 QM1 的薪资查询应用程序，该应用程序将薪资查询消息发送到在队列管理器 QM2 上运行的薪资处理应用程序。 工资单查询应用程序需要对其发送回 QM1 的查询的应答。

关于此任务

此示例说明了允许应用程序使用队列管理器 QM1 将消息放在队列管理器 QM2 上的队列上所需的准备工作。 在 QM2 上运行的应用程序可以检索这些消息，并将响应发送到 QM1 上的应答队列。

此示例说明如何使用 TCP/IP 连接。 此示例假定将触发通道以在第一条消息到达他们正在服务的传输队列时启动。

此示例使用 SYSTEM.CHANNEL.INITQ 作为启动队列。 此队列已由 IBM MQ 定义。 您可以使用不同的启动队列，但必须自行定义该队列，使用 STRMQMCHLI 命令启动通道启动程序的新实例，并为其提供启动队列的名称。 有关触发通道的更多信息，请参阅 触发通道。

注: CD MQ Adv. 使用 TCP/IP 的信息通道可以指向 IBM Aspera faspio Gateway，它提供了一个快速的 TCP/IP 隧道，可以显著提高网络吞吐量。 请参阅在 Linux 或 Windows 上定义 Aspera gateway 连接。

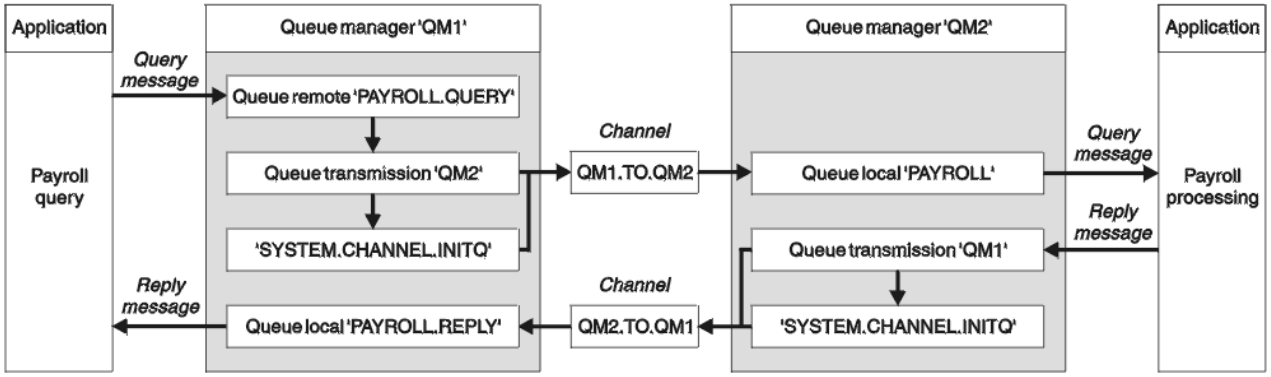

图 *8: IBM MQ for IBM i* 的消息通道示例

薪资查询消息在名为 QM1.TO.QM2 的发送方/接收方通道上从 QM1 发送到 QM2 ，应答消息在另一个名为 QM2.TO.QM1 的发送方/接收方通道上从 QM2 发送回 QM1 。 一旦这两个通道有消息要发送到另一个队列管理器，就会触发这两个通道启动。

工资单查询应用程序将查询消息放入 QM1 上定义的远程队列 “PAYROLL.QUERY” 。 此远程队列定义解析为 QM2 上的本地队列 “PAYROLL” 。 此外，工资单查询应用程序指定将对查询的应答发送到 QM1 上的本地队列 “PAYROLL.REPLY” 。 工资单处理应用程序从 QM2 上的本地队列 “工资单” 获取消息，并将应答发送到所需的任何位置; 在此情况下，本地队列 “PAYROLL.REPLY” 位于 QM1 上。

假定这两个队列管理器都在 IBM i 上运行。 在示例定义中， QM1 的主机地址为 192.0.2.0 ，并且正在侦听端口 1411。 QM2 具有主机地址 192.0.2.1 ，并且正在侦听端口 1412。 此示例假定这些队列管理器已在 IBM i 系统上定义，并且可供使用。

需要在 QM1 上创建的对象定义包括:

- 远程队列定义， PAYROLL.QUERY
- 传输队列定义， QM2 (缺省值 = 远程队列管理器名称)
- 发送方通道定义， QM1.TO.QM2
- 接收方通道定义， QM2.TO.QM1
- 应答队列定义 PAYROLL.REPLY

需要在 QM2 上创建的对象定义包括:

- 本地队列定义， PAYROLL
- 传输队列定义， QM1 (缺省值 = 远程队列管理器名称)
- 发送方通道定义， QM2.TO.QM1
- 接收方通道定义， QM1.TO.QM2

连接详细信息在发送方通道定义的 CONNAME 属性中提供。

您可以在 第 134 页的图 8 中查看排列图。

过程

请参阅:

- 第 135 页的『在 IBM i 上设置消息通道代理程序』 以获取有关设置消息通道的详细信息
- 第 137 页的『运行和展开 IBM i 的示例』 ，以获取有关如何连接更多应用程序和用户出口的建议。

IBM i 在 IBM i 上设置消息通道代理程序

以下对象定义允许连接到队列管理器 QM1 的应用程序向 QM2 上名为 PAYROLL 的队列发送请求消息，在 QM1 上接收名为 PAYROLL.REPLY 的队列上的应答，允许连接到队列管理器 QM2 的应用程序从名为 PAYROLL 的本地队列检索请求消息，并将这些请求消息的应答放入队列管理器 QM1 上名为 PAYROLL.REPLY 的队列。

关于此任务

已为所有对象定义提供了 TEXT 属性。 提供的其他属性是使示例工作所需的最小值。 未提供的属性采用队列管理器 QM1 和 QM2 的缺省值。

您不需要提供远程队列定义以允许将应答返回到 QM1。 从本地队列 PAYROLL 中检索的消息的消息描述符同时包含应答队列和应答队列管理器名称。 因此，只要 QM2 可以将应答队列管理器名称解析为队列管理器 QM2 上传输队列的名称，就可以发送应答消息。 在此示例中，应答队列管理器名称为 QM1 ，因此队列管理器 QM2 需要同名的传输队列。

过程

- 在队列管理器 QM1:

  a) 通过使用带有以下属性的 CRTMQMQ 命令来设置远程队列定义:

QNAME	'PAYROLL.QUERY'
QTYPE	*RMT
文本	' QM2 的远程队列'
PUTENBL	*YES
TMQNAME	"QM2" (缺省值为远程队列管理器名称)
RMTQNAME	'工资单'
RMTMQMNAME	'QM2'

  注: 远程队列定义不是物理队列，而是将消息定向到传输队列 QM2 的一种方法，以便可以将这些消息发送到队列管理器 QM2。

  b) 使用带有以下属性的 CRTMQMQ 命令来设置传输队列定义:

QNAME	QM2
QTYPE	*LCL
文本	'将队列传输到 QM2'
USAGE	*TMQ
PUTENBL	*YES
GETENBL	*YES
TRGENBL	*YES
TRGTYPE	*FIRST
INITQNAME	SYSTEM.CHANNEL.INITQ
TRIGDATA	QM1.TO.QM2

  将第一条消息放入此传输队列时，会将触发器消息发送到启动队列 SYSTEM.CHANNEL.INITQ。 通道启动程序从启动队列中获取消息，并启动在指定进程中标识的通道。

  c) 使用带有以下属性的 CRTMQCHL 命令来设置发送方通道定义:

CHLNAME	QM1.TO.QM2

CHLTYPE	*SDR
TRPTYPE	*TCP
文本	"发送方通道到 QM2"
TMQNAME	QM2
CONNAME	'192.0.2.1(1412)'

d) 通过使用带有以下属性的 CRTMQCHL 命令来设置接收方通道定义:

CHLNAME	QM2.TO.QM1
CHLTYPE	*RCVR
TRPTYPE	*TCP
文本	"来自 QM2 的接收方通道"

e) 使用带有以下属性的 CRTMQMQ 命令来设置应答队列定义:

QNAME	PAYROLL.REPLY
QTYPE	*LCL
文本	"针对发送到 QM2 的查询消息的应答的应答队列"
PUTENBL	*YES
GETENBL	*YES

应答队列定义为 PUT (ENABLED)。 此定义确保可以将应答消息放入队列。 如果无法将应答放入应答队列，那么会将其发送至 QM1 上的死信队列，如果此队列不可用，请将其保留在队列管理器 QM2 上的传输队列 QM1 上。 队列已定义为 GET (ENABLED) ，以允许检索应答消息。

- 在队列管理器 QM2:

a) 使用带有以下属性的 CRTMQMQ 命令来设置本地队列定义:

QNAME	工资单
QTYPE	*LCL
文本	" QM1 工资单详细信息的本地队列"
PUTENBL	*YES
GETENBL	*YES

由于与队列管理器 QM1 上的应答队列定义相同的原因，此队列定义为 PUT (ENABLED) 和 GET (ENABLED)。

b) 使用带有以下属性的 CRTMQMQ 命令来设置传输队列定义:

QNAME	QM1
QTYPE	*LCL
文本	'将队列传输到 QM1'
USAGE	*TMQ
PUTENBL	*YES
GETENBL	*YES
TRGENBL	*YES
TRGTYPE	*FIRST
INITQNAME	SYSTEM.CHANNEL.INITQ

TRIGDATA	QM2.TO.QM1

将第一条消息放入此传输队列时，会将触发器消息发送到启动队列 SYSTEM.CHANNEL.INITQ。 通道启动程序从启动队列获取消息并启动触发器数据中标识的通道。

c) 通过使用带有以下属性的 CRTMQMCHL 命令来设置发送方通道定义:

CHLNAME	QM2.TO.QM1
CHLTYPE	*SDR
TRPTYPE	*TCP
文本	"发送方通道到 QM1"
TMQNAME	QM1
CONNAME	'192.0.2.0(1411)'

d) 使用带有以下属性的 CRTMQMCHL 命令来设置接收方通道定义:

CHLNAME	QM1.TO.QM2
CHLTYPE	*RCVR
TRPTYPE	*TCP
文本	'来自 QM1 的接收方通道'

IBM i 运行和展开 IBM i 的示例

有关启动通道启动程序和侦听器的信息以及有关在此场景中展开的建议。

关于此任务

一旦创建了这些定义，您就需要:

- 在每个队列管理器上启动通道启动程序。
- 启动每个队列管理器的侦听器。

然后，应用程序可以相互发送消息。 通道由到达每个传输队列的第一条消息触发启动，因此您不需要发出 STRMQMCHL 命令。

您还可以展开该示例。

过程

1. 启动通道启动程序和侦听器。
   请参阅 在 IBM i 上监视和控制通道 ，以获取有关启动通道启动程序和侦听器的详细信息。
2. 您可以通过以下方法展开此示例:
   - 添加更多队列和通道定义以允许其他应用程序在两个队列管理器之间发送消息。
   - 在通道上添加用户出口程序以允许进行链路加密，安全性检查或其他消息处理。
   - 使用队列管理器别名和应答队列别名来了解有关如何在队列管理器网络的组织中使用这些对象的更多信息。

   z/OS 有关此示例使用 MQSC 命令的版本，请参阅 第 138 页的『示例: 在 z/OS 上规划消息通道』。

z/OS 示例: 在 z/OS 上规划消息通道

如何将 z/OS 或 MVS 队列管理器连接在一起，以便可以在它们之间发送消息。 此示例涉及连接到队列管理器 QM1 的薪资查询应用程序，该应用程序将薪资查询消息发送到在队列管理器 QM2 上运行的薪资处理应用程序。 工资单查询应用程序需要对其发送回 QM1 的查询的应答。

关于此任务

此示例说明了允许应用程序使用队列管理器 QM1 将消息放在队列管理器 QM2 上的队列上所需的准备工作。 在 QM2 上运行的应用程序可以检索这些消息，并将响应发送到 QM1 上的应答队列。

此示例说明如何使用 TCP/IP 和 LU 6.2 连接。 此示例假定将触发通道以在第一条消息到达他们正在服务的传输队列时启动。

注: MQ Adv. 使用 TCP/IP 的信息通道可以指向 IBM Aspera faspio Gateway，它提供了一个快速的 TCP/IP 隧道，可以显著提高网络吞吐量。 请参阅在 Linux 或 Windows 上定义 Aspera gateway 连接。

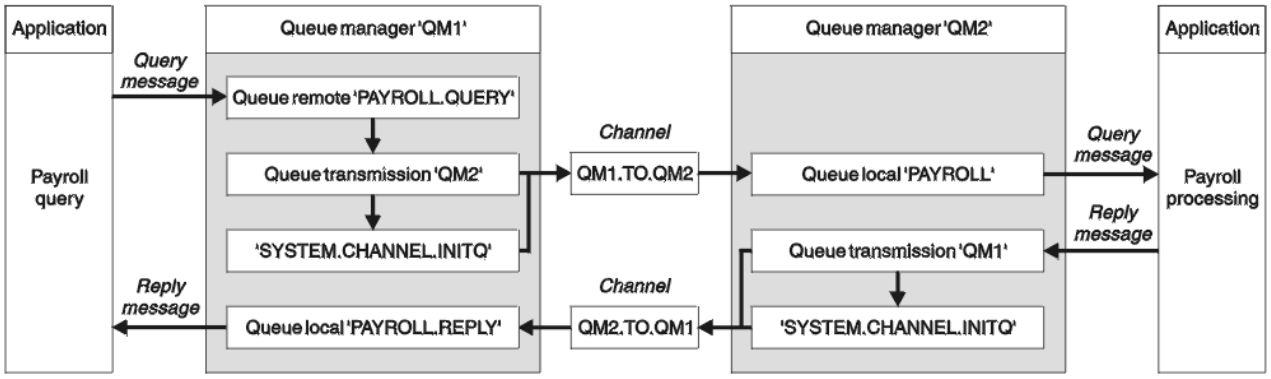

图 *9: IBM MQ for z/OS* 的第一个示例

薪资查询消息在名为 QM1.TO.QM2 的发送方/接收方通道上从 QM1 发送到 QM2 ，应答消息在另一个名为 QM2.TO.QM1 的发送方/接收方通道上从 QM2 发送回 QM1 。 一旦这两个通道有消息要发送到另一个队列管理器，就会触发这两个通道启动。

工资单查询应用程序将查询消息放入 QM1 上定义的远程队列 “PAYROLL.QUERY” 。 此远程队列定义解析为 QM2 上的本地队列 “PAYROLL” 。 此外，工资单查询应用程序指定将对查询的应答发送到 QM1 上的本地队列 “PAYROLL.REPLY” 。 工资单处理应用程序从 QM2 上的本地队列 “工资单” 获取消息，并将应答发送到所需的任何位置; 在此情况下，本地队列 “PAYROLL.REPLY” 位于 QM1 上。

假定这两个队列管理器都在 z/OS 上运行。 在 TCP/IP 的示例定义中， QM1 的主机地址为 192.0.2.0 ，并且正在侦听端口 1411 ，而 QM2 的主机地址为 192.0.2.1 ，并且正在侦听端口 1412。 在 LU 6.2 的定义中，QM1 正在侦听名为 LUNAME1 的符号 LUN ， QM2 正在侦听名为 LUNAME2 的符号 LUN。 此示例假定这些 LUN 已在 z/OS 系统上定义并且可供使用。 要定义它们，请参阅 第 41 页的『示例: 在 z/OS 上设置 IBM MQ 跨平台通信』。

需要在 QM1 上创建的对象定义包括:

- 远程队列定义， PAYROLL.QUERY
- 传输队列定义， QM2 (缺省值 = 远程队列管理器名称)
- 发送方通道定义， QM1.TO.QM2
- 接收方通道定义， QM2.TO.QM1
- 应答队列定义 PAYROLL.REPLY

需要在 QM2 上创建的对象定义包括:

- 本地队列定义， PAYROLL
- 传输队列定义， QM1 (缺省值 = 远程队列管理器名称)
- 发送方通道定义， QM2.TO.QM1

- 接收方通道定义， QM1.TO.QM2

此示例假定所有 SYSTEM.COMMAND.* 和 SYSTEM.CHANNEL.* 已定义运行 DQM 所需的队列，如提供的样本定义 **CSQ4INSG** 和 **CSQ4INSX** 中所示。

连接详细信息在发送方通道定义的 CONNAME 属性中提供。

您可以在 第 138 页的图 9 中查看排列图。

过程

请参阅：

- 第 139 页的『在 z/OS 上设置消息通道代理程序』 以获取有关设置消息通道的详细信息
- 第 141 页的『Running and expanding the example for z/OS』，以获取有关如何连接更多应用程序和用户出口的建议。

z/OS 在 z/OS 上设置消息通道代理程序

以下对象定义允许连接到队列管理器 QM1 的应用程序将请求消息发送到 QM2 上的名为 PAYROLL 的队列，还允许应用程序在 QM1 上的名为 PAYROLL.REPLY 的队列上接收应答。 这些定义还允许连接到队列管理器 QM2 的应用程序从名为 PAYROLL 的本地队列检索请求消息，并将对这些请求消息的应答放入队列管理器 QM1 上名为 PAYROLL.REPLY 的队列。

关于此任务

所有对象定义都已提供 DESCR 和 REPLACE 属性，并且是使示例工作所需的最低要求。 未提供的属性采用队列管理器 QM1 和 QM2 的缺省值。

您不需要提供远程队列定义以允许将应答返回到 QM1。 从本地队列 PAYROLL 中检索的消息的消息描述符同时包含应答队列和应答队列管理器名称。 因此，只要 QM2 可以将应答队列管理器名称解析为队列管理器 QM2 上传输队列的名称，就可以发送应答消息。 在此示例中，应答队列管理器名称为 QM1，因此队列管理器 QM2 需要同名的传输队列。

过程

- 在队列管理器 QM1：

  a) 设置远程队列定义：

  ```
  DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QM2') REPLACE +
  PUT(ENABLED) XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
  ```

  注： 远程队列定义不是物理队列，而是将消息定向到传输队列 QM2 的一种方法，以便可以将这些消息发送到队列管理器 QM2。

  b) 设置传输队列定义：

  ```
  DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') REPLACE +
  USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
  TRIGDATA(QM1.TO.QM2) INITQ(SYSTEM.CHANNEL.INITQ)
  ```

  将第一条消息放入此传输队列时，会将触发器消息发送到启动队列 SYSTEM.CHANNEL.INITQ。 通道启动程序从启动队列获取消息并启动触发器数据中标识的通道。 通道启动程序只能从 SYSTEM.CHANNEL.INITQ 队列，因此请勿使用任何其他队列作为启动队列。

  c) 设置发送方通道定义：

  对于 TCP/IP 连接：

  ```
  DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) +
  REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +
  CONNAME('192.0.2.1(1412)')
  ```

  对于 LU 6.2 连接：

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +
CONNAME('LUNAME2')
```

d) 设置接收方通道定义:

对于 TCP/IP 连接:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QM2')
```

对于 LU 6.2 连接:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QM2')
```

e) 设置应答队列定义:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to query messages sent to QM2')
```

应答队列定义为 PUT (ENABLED) ，可确保应答消息可放入队列。 如果无法将应答放入应答队列，那么会将其发送至 QM1 上的死信队列，如果此队列不可用，请将其保留在队列管理器 QM2 上的传输队列 QM1 上。 队列已定义为 GET (ENABLED) ，以允许检索应答消息。

- 在队列管理器 QM2:

a) 设置本地队列定义:

```
DEFINE QLOCAL(PAYROLL) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Local queue for QM1 payroll details')
```

由于与队列管理器 QM1 上的应答队列定义相同的原因，此队列定义为 PUT (ENABLED) 和 GET (ENABLED)。

b) 设置传输队列定义:

```
DEFINE QLOCAL(QM1) DESCR('Transmission queue to QM1') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QM2.TO.QM1) INITQ(SYSTEM.CHANNEL.INITQ)
```

将第一条消息放入此传输队列时，会将触发器消息发送到启动队列 SYSTEM.CHANNEL.INITQ。 通道启动程序从启动队列获取消息并启动触发器数据中标识的通道。 通道启动程序只能从 SYSTEM.CHANNEL.INITQ 因此请勿将任何其他队列用作启动队列。

c) 设置发送方通道定义:

对于 TCP/IP 连接:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +
CONNAME('192.0.2.0(1411)')
```

对于 LU 6.2 连接:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +
CONNAME('LUNAME1')
```

d) 设置接收方通道定义:

对于 TCP/IP 连接:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QM1')
```

对于 LU 6.2 连接:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QM1')
```

z/OS Running and expanding the example for z/OS

Information about starting the channel initiator and listener and suggestions for expanding on this example.

About this task

Once these definitions have been created, you need to:

- Start the channel initiator on each queue manager.
- Start the listener for each queue manager.

The applications can then send messages to each other. Because the channels are triggered to start by the arrival of the first message on each transmission queue, you do not need to issue the START CHANNEL MQSC command.

You can also expand the example.

Procedure

1. Start the channel initiator and listener.

   See Starting a channel initiator, and Starting a channel listener for details on how to start a channel initiator and listener.
2. You can expand this example by:
   - Adding more queues, and channel definitions to allow other applications to send messages between the two queue managers.
   - Adding user exit programs on the channels to allow for link encryption, security checking, or additional message processing.
   - Using queue manager aliases and reply-to queue aliases to understand more about how these aliases can be used in the organization of your queue manager network.

z/OS Example: planning a message channel for z/OS using queue sharing groups

This example illustrates the preparations needed to allow an application using queue manager QM3 to put a message on a queue in a queue sharing group that has queue members QM4 and QM5, and also shows the IBM MQ commands (MQSC) that you can use in IBM MQ for z/OS for distributed queuing with queue sharing groups.

About this task

Ensure you are familiar with the example in “示例: 在 z/OS 上规划消息通道” on page 138 before trying this one. This example expands the payroll query scenario of that example, to show how to add higher availability of query processing by adding more serving applications to serve a shared queue.

The payroll query application is now connected to queue manager QM3 and puts a query to the remote queue 'PAYROLL QUERY' defined on QM3. This remote queue definition resolves to the shared queue 'PAYROLL' hosted by the queue managers in the queue sharing group QSG1. The payroll processing application now has two instances running, one connected to QM4 and one connected to QM5.

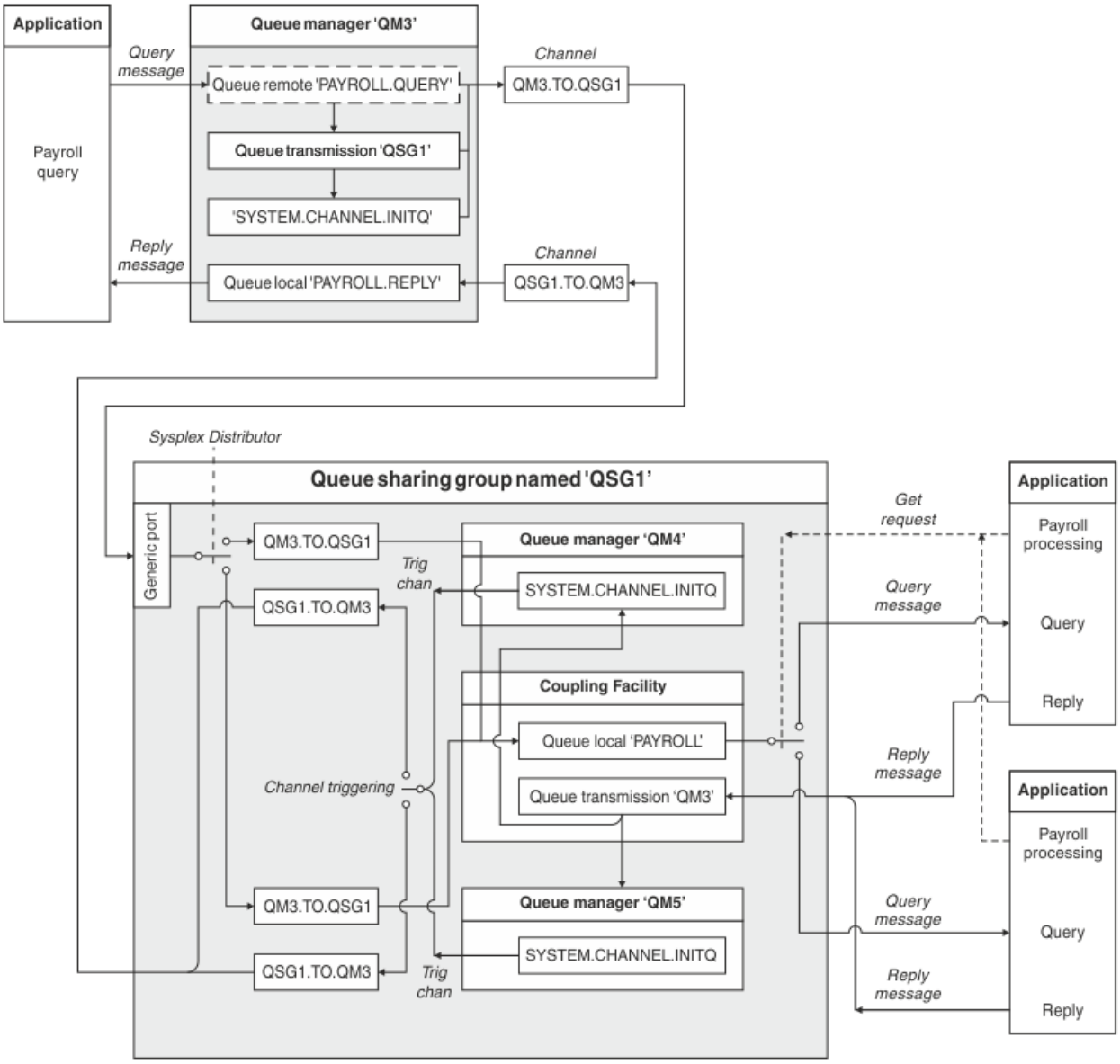

Figure 10. Message channel planning example for IBM MQ for z/OS using queue sharing groups

All three queue managers are assumed to be running on z/OS. In the example definitions for TCP/IP, QM4 has a VIPA address of MVSIP01 and QM5 has a VIPA address of MVSIP02. Both queue managers are listening on port 1414. The generic address that Sysplex Distributor provides for this group is QSG1.MVSIP. QM3 has a host address of 192.0.2.0 and is listening on port 1411.

In the example definitions for LU6.2, QM3 is listening on a symbolic luname called LUNAME1. The name of the generic resource defined for VTAM for the lunames listened on by QM4 and QM5 is LUQSG1. The example assumes that they are already defined on your z/OS system and are available for use. To define them see “在队列共享组中建立 LU 6.2 连接” on page 45.

In this example QSG1 is the name of a queue sharing group, and queue managers QM4 and QM5 are the names of members of the group.

Procedure

See:

- “设置队列共享组定义和不在队列共享组中的队列管理器 QM3” on page 143 for details on setting up the definitions.

- “运行 z/OS 的队列共享组示例” on page 144 for details on starting the channel initiators and listeners for each queue manager.

z/OS 设置队列共享组定义和不在队列共享组中的队列管理器 QM3

为队列共享组的一个成员生成以下对象定义使它们可供所有其他成员使用。 QM3 不是队列共享组的成员。

关于此任务

队列管理器 QM4 和 QM5 是队列共享组的成员。 为 QM4 生成的定义也可用于 QM5。

假定耦合设施列表结构称为 "APPLICATION1"。 如果它不称为 "APPLICATION1" ，那么必须使用您自己的耦合设施列表结构名称作为示例。

由于 QM3 不是队列共享组的成员，因此您需要该队列管理器的对象定义以允许其将消息放入队列共享组中的队列。

过程

- 设置队列共享组定义的共享对象:
  a) 使用以下命令来设置存储在 Db2 中的共享对象定义及其存储在耦合设施中的关联消息。

```
DEFINE QLOCAL(PAYROLL) QSGDISP(SHARED) REPLACE PUT(ENABLED) GET(ENABLED) +
CFSTRUCT(APPLICATION1) +
DESCR('Shared queue for payroll details')

DEFINE QLOCAL(QM3) QSGDISP(SHARED) REPLACE USAGE(XMITQ) PUT(ENABLED) +
CFSTRUCT(APPLICATION1) +
DESCR('Transmission queue to QM3') TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QSG1.TO.QM3) GET(ENABLED) INITQ(SYSTEM.CHANNEL.INITQ)
```

- 使用以下命令来设置存储在 Db2 中的组对象定义。
  队列共享组中的每个队列管理器都将创建已定义对象的本地副本。
  a) 设置发送方通道:

  TCP/IP 连接的发送方通道定义:

```
DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(SDR) QSGDISP(GROUP) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM3') XMITQ(QM3) +
CONNAME('192.0.2.0(1411)')
```

  LU 6.2 连接的发送方通道定义:

```
DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(SDR) QSGDISP(GROUP) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QM3') XMITQ(QM3) +
CONNAME('LUNAME1')
```

  b) 设置接收方通道:

  TCP/IP 连接的接收方通道定义:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QM3') QSGDISP(GROUP)
```

  LU 6.2 连接的接收方通道定义:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QM3') QSGDISP(GROUP)
```

- 设置队列管理器 QM3 对象定义。
  a) 设置 CONNAME

  此通道的 CONNAME 是队列共享组的通用地址，根据传输类型而有所不同。

  对于 TCP/IP 连接:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QSG1') XMITQ(QSG1) +
CONNAME('QSG1.MVSIP(1414)')
```

对于 LU 6.2 连接:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(SDR) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QSG1') XMITQ(QSG1) +
CONNAME('LUQSG1') TPNAME('MQSERIES') MODENAME('#INTER')
```

b) 设置其他定义。

这些定义需要与 第 138 页的『示例: 在 z/OS 上规划消息通道』的子主题中使用的定义相同。

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QSG1') REPLACE +
PUT(ENABLED) XMITQ(QSG1) RNAME(APPL) RQMNAME(QSG1)

DEFINE QLOCAL(QSG1) DESCR('Transmission queue to QSG1') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QM3.TO.QSG1) INITQ(SYSTEM.CHANNEL.INITQ)

DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QSG1')

DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QSG1')

DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to query messages sent to QSG1')
```

z/OS 运行 z/OS 的队列共享组示例

有关启动通道启动程序和侦听器的信息。

关于此任务

创建所需对象后，需要执行以下操作:

- 启动所有三个队列管理器的通道启动程序。
- 启动这两个队列管理器的侦听器。

过程

1. 启动通道启动程序。

   有关如何启动通道启动程序的详细信息，请参阅 启动通道启动程序 。
2. 启动侦听器。

   有关如何启动侦听器的详细信息，请参阅 启动通道侦听器 。

   对于 TCP/IP 连接，组的每个成员都必须具有正在侦听端口 1414 的组侦听器。

   ```
   STA LSTR PORT(1414) IPADDR(MVSIP01) INDISP(GROUP)
   ```

   例如，前一个条目在 QM4 上启动侦听器。

   对于 LU6.2 连接，组的每个成员都必须启动一个正在侦听符号 luname 的组侦听器。 此 luname 必须对应于通用资源 LUQSG1。

   ```
   STA LSTR PORT(1411)
   ```

   前一个条目在 QM3 上启动侦听器。

使用别名来引用 MQ 库

您可以定义别名以引用 JCL 中的 MQ 库，而不是直接使用 MQ 库的名称。 然后，如果 MQ 库的名称发生更改，那么您只能删除并重新定义别名。

示例

以下示例定义别名 MQM.SCSQANLE 以引用 MQ 库 MQM.V600.SCSQANLE:

```
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE (MQM.SCSQANLE)
DEFINE ALIAS (NAME(MQM.SCSQANLE) RELATE(MQM.V600.SCSQANLE))
/*
```

然后，引用 MQM.V600.SCSQANLE 库使用别名 MQM.SCSQANLE。

注: 库名和别名必须位于同一目录中，因此请对两者使用相同的高级限定符; 在此示例中，高级限定符为 MQM。

Managed File Transfer 配置参考信息

用于帮助您配置 Managed File Transfer 的参考信息。

在 MFT 属性中使用环境变量

可以在表示文件或目录位置的 Managed File Transfer 属性中使用环境变量。 这允许在运行产品部分时使用的文件或目录的位置因当前环境而异 (例如，运行命令的用户)。

以下属性可接受文件或目录位置，因此可以包含环境变量:

- agentQMgrAuthenticationCredentialsFile
- agentSslKeyStore
- agentSslKeyStoreCredentialsFile
- agentSslTrustStore
- agentSslTrustStoreCredentialsFile
- cdNodeKeystoreCredentialsFile
- cdNodeTruststoreCredentialsFile
- cdTmpDir
- cdNodeKeystore
- cdNodeTruststore
- commandPath
- connectionQMgrAuthenticationCredentialsFile
- connectionSslKeyStore
- connectionSslKeyStoreCredentialsFile
- connectionSslTrustStore
- connectionSslTrustStoreCredentialsFile
- coordinationSslKeyStore
- coordinationSslKeyStoreCredentialsFile
- coordinationQMgrAuthenticationCredentialsFile
- coordinationSslTrustStore
- coordinationSslTrustStoreCredentialsFile
- exitClassPath
- exitNativeLibraryPath
- javaCoreTriggerFile
- loggerQMgrAuthenticationCredentialsFile
- sandboxRoot

- transferRoot
- wmqfte.database.credentials.file

Windows 的示例

Windows 在 Windows 系统上的此示例中，用户 fteuser 使用环境变量 USERPROFILE:

```
wmqfte.database.credentials.file=%USERPROFILE%\\logger\\mqmftcredentials.xml
```

解析为以下文件路径:

```
C:\Users\fteuser\logger\mqmftcredentials.xml
```

AIX and Linux 的示例

Linux AIX 本示例表示在 UNIX 系统上，用户 fteuser 使用环境变量 HOME:

```
transferRoot=$HOME/fte/
```

解析为以下文件路径:

```
/home/fteuser/fte/
```

相关参考

第 167 页的『MFT coordination.properties 文件』
coordination.properties 文件指定到协调队列管理器的连接详细信息。 由于一些 Managed File Transfer 安装可能共享同一协调队列管理器，因此您可以将符号链接用于共享驱动器上的通用 coordination.properties 文件。

第 170 页的『MFT command.properties 文件』
command.properties 文件指定发出命令时要连接的命令队列管理器以及 Managed File Transfer 联系该队列管理器所需的信息。

第 149 页的『MFT agent.properties 文件』
每个 Managed File Transfer Agent 都有自己的属性文件 agent.properties，其中必须包含代理用于连接到其队列管理器的信息。 agent.properties 文件还可以包含更改代理行为的属性。

MFT 的 SSL/TLS 属性

第 173 页的『MFT logger.properties 文件』
Managed File Transfer 记录器具有一组配置属性。 在 logger.properties 文件中指定这些属性，该文件位于 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name* 目录中。

用户出口的 MFT 代理属性

协议网桥属性文件格式

Connect:Direct 进程定义文件格式

Connect:Direct 节点属性文件格式

MFT installation.properties 文件

installation.properties 文件指定一组缺省配置选项的名称。 该条目将 Managed File Transfer 指向包含要使用的配置的一组结构化目录和属性文件。 通常，一组配置选项的名称是关联的协调队列管理器的名称。

该文件由安装程序创建，可通过使用 **fteChangeDefaultConfigurationOptions** 命令进行更改。

installation.properties 文件位于您的 *MQ_DATA_PATH* 目录中。 例如，在 Windows 上，缺省文件位置为 *MQ_DATA_PATH*\mqft\installations*installation_name* ，而在 AIX and Linux 系统上，缺省文件位置为 /var/mqm/mqft/installations/*installation_name*。

对于 Redistributable Managed File Transfer Agent，将在您运行 **fteCreateEnvironment** 命令时设置数据路径。 如果您运行此命令并使用 **-d** 参数指定您选择的位置，那么将为此位置设置数据路径。 如果未使用

fteCreateEnvironment 命令指定位置，那么将在解压缩 Redistributable Managed File Transfer Agent 的根目录下创建目录 `mftdata` 。 Redistributable Managed File Transfer Agent 的 `installation.properties` 文件位于 `MQ_DATA_PATH\mqft\installations\MFTZipInstall` 目录中。

`installation.properties` 文件包含以下值：

表 *64:* 基本属性

属性名	描述	缺省值
commandMessagePriority	为 **fteStopAgent**、**fteCancelTransfer** 和 **ftePingAgent** 命令的内部消息和命令消息设置优先级。 例如，如果您提交大量传输请求以快速连续传输小文件，那么新传输请求会在源代理的命令队列中排队。 外部和内部消息具有缺省 IBM MQ 消息优先级，因此内部消息会被新传输请求阻止。 这可能导致超出传输协商时间以及恢复传输。 您还可以使用 commandMessagePriority 属性设置内部确认和期望确认消息的优先级。 要将内部 Managed File Transfer 消息的优先级设置为高于新的传输请求，请将该属性设置为 1（最低值）到 9（最高值）之间的值。 commandMessagePriority 属性的默认值为 8。 这意味着，如果代理命令队列上的 IBM MQ 属性 DEFPRTY (缺省优先级) 小于或等于 7 ，那么将在新的传输请求之前对内部协商消息进行优先级划分。 如果将 DEFPRTY 属性的值设置为 8 或 9 以保持 commandMessagePriority 属性的有效性，必须更改 DEFPRTY 或 commandMessagePriority 属性。	缺省值为 8。
commonCredentialsKeyFile	包含加密凭证时使用的凭证密钥的文件的标准路径名。 MFT 凭证文件的最常用名称为 `MQMFTCredentials.xml`。 有关使用 commonCredentialsKeyFile 属性的更多信息，请参阅凭证解密。	密钥文件的标准路径
defaultProperties	缺省配置选项集的名称。 该值是位于配置目录中的目录名称，它包含指定配置信息的目录和属性文件。	没有缺省值

表 64: 基本属性 (继续)		
属性名	**描述**	**缺省值**
enableFunctionalFixPack	启用修订包功能级别。 缺省情况下，不会启用修订包中的任何新功能。 将该属性设置为版本标识，以启用该版本提供的新功能。 可以指定带有或不带句点字符 (.) 的版本标识。 例如，要使用 IBM MQ 8.0.0 Fix Pack 2 提供的函数，请将此属性设置为 8002 或 8.0.0.2。	没有缺省值
messagePublicationFormat	允许您指定 MFT 代理对其状态 XML 消息使用的消息发布格式。 可以将该属性设置为以下值: **messagePublicationFormat=mixed** 在不使用 MQMD FORMAT (MQFMT_NONE) 的情况下发布消息，在 /LOG 主题树下发布（采用 MQMD 格式 MQFMT_STRING 发布）的消息除外。 **messagePublicationFormat=MQFMT_NONE** 在不使用 MQMD FORMAT 的情况下发布消息。 **messagePublicationFormat=MQFMT_STRING** 采用字符串格式发布消息。	messagePublicationFormat=mixed
z/OS **特定于 z/OS:**		
productId	要对其记录 MFT 用法的产品类型: • 独立 Managed File Transfer 产品。 （MFT 为 **productID**）。 • IBM MQ Advanced 产品的一部分。 （ADVANCED 为 **productID**）。 • z/OS IBM MQ Advanced for z/OS Value Unit Edition 产品的一部分。 （ADVANCEDVUE 为 **productID**）。 请参阅 报告产品信息，以获取有关产品使用情况记录的更多信息。 Multi 在多平台上忽略该属性。	MFT

以下文本是 `installation.properties` 文件的内容示例。

```
defaultProperties=ERIS
```

ERIS 是一个目录名称，与 installation.properties 文件位于同一个目录中。 目录 ERIS 包含描述一组配置选项的目录和属性文件。

相关概念

多平台上的 MFT 配置选项

相关参考

fteChangeDefaultConfigurationOptions

MFT agent.properties 文件

每个 Managed File Transfer Agent 都有自己的属性文件 agent.properties，其中必须包含代理用于连接到其队列管理器的信息。 agent.properties 文件还可以包含更改代理行为的属性。

agent.properties 文件由安装程序或 **fteCreateAgent**， **fteCreateBridgeAgent** 或 **fteCreateCDAgent** 命令创建。 您可以将任意这些命令与 **-f** 标志一起使用，以更改基本代理队列管理器属性和那些与创建的代理类型关联的高级代理属性。 要更改或添加高级代理属性，您必须在文本编辑器中编辑此文件。

Multi 在 Multiplatforms 版上，代理程序的 agent.properties 文件位于 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/agents/*agent_name* 目录中。

z/OS 在 z/OS 上， agent.properties 文件位置为 $BFG_CONFIG variable/mqft/config/coordination_qmgr_name/agents/agent_name。

如果更改 agent.properties 文件，那么必须重新启动代理以获取更改。

可以在表示文件或目录位置的某些 Managed File Transfer 属性中使用环境变量。 这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化，例如哪个用户在运行进程。 有关更多信息，请参阅 第 145 页的『在 MFT 属性中使用环境变量』。

Windows

注: 在 Windows 上，有两个属性:

- windowsService
- windowsServiceVersion

由 MFT 命令添加到 agent.properties 文件中，这些命令用于设置要作为 Windows 服务运行的代理程序。

您不应手动添加或修改这些属性，因为这将阻止代理程序正常工作。

有关如何设置代理以作为 Windows 服务，请参见开始 MFT 代理人作为 Windows 服务。

基本代理程序属性

每个 MFT agent.properties 文件都包含以下基本代理程序属性:

表 *65:* 基本代理程序属性

属性名	描述	缺省值
agentName	代理的名称。 代理的名称必须符合 IBM MQ 对象命名约定。 有关更多信息，请参阅 第 197 页的『MFT 对象命名约定』。	没有缺省值
agentDesc	代理的描述 - 如果您选择创建描述。	没有缺省值
agentQMgr	代理队列管理器名称。	没有缺省值
agentQMgrHost	代理队列管理器的主机名或 IP 地址。	没有缺省值
agentQMgrPort	与代理队列管理器建立的客户机连接所使用的端口号。	1414
agentQMgrChannel	用于与代理队列管理器连接的 SVRCONN 通道名称。	SYSTEM.DEF.SVRCONN

表 65: 基本代理程序属性 (继续)		
属性名	**描述**	**缺省值**
agentType	代理类型: • 标准非网桥代理 (STANDARD) • 协议网桥代理 (BRIDGE) • Connect:Direct 网桥代理 (CD_BRIDGE) • IBM Integration Bus 使用的嵌入式代理 (EMBEDDED) • Sterling File Gateway 嵌入式代理 (SFG)	标准

如果没有为 **agentQMgrHost** 属性指定值，则默认使用绑定模式。

如果为 **agentQMgrHost** 属性指定了值，但没有为 **agentQMgrPort** 和 **agentQMgrChannel** 属性指定值，则默认使用 1414 端口号和 SYSTEM.DEF.SVRCONN 频道。

高级代理属性

Managed File Transfer 还提供帮助您配置代理的更多高级代理属性。 如果您要使用以下任意属性，请手动编辑 `agent.properties` 文件以添加所需的高级属性。括号、逗号 (,) 和反斜线 (\) 是 MFT 命令中的特殊字符，必须用反斜线 (\) 字符转义。 Windows Windows 上的文件路径可以使用双反斜线 (\\) 作为分隔符，也可以使用单个正斜线 (/) 指定。 有关 属性文件中字符转义的更多信息，请参阅 文档中 Java Oracle Properties 类的 Javadoc。

- 表 2. 高级代理属性代理大小
- 表 3. 高级代理属性：代码页面
- 表 4. 高级代理属性：命令
- 表 5. 高级代理属性连接
- 表 6. 高级代理属性： Connect:Direct 网桥
- 表 7. 高级代理属性：文件到信息和信息到文件
- 表 8. 高级代理属性：常规
- 表 9. 先进的代理属性高可用性
- 表 10. 高级代理属性：输入/输出
- 表 11. 高级代理属性：传输日志
- 表 12. 先进的代理属性：多渠道支持
- 表 13. 高级代理属性：多实例
- 表 14. 高级代理属性：流程控制器
- 表 15. 高级代理属性：协议桥接
- 表 16. 高级代理属性协议桥代理日志
- 表 17. 高级代理属性队列
- 表 18. 高级代理属性：资源监控
- 表 19. 高级代理属性根目录
- 表 20. 高级代理属性调度程序属性
- 表 21. 高级代理属性安全性
- 表 22. 高级代理属性：SSL/TLS
- 表 24. 高级代理属性超时
- 表 25. 高级代理属性跟踪和记录
- 表 26. 高级代理属性：传输限制
- 表 27. 高级代理属性：传输恢复超时
- 表 28. 高级代理属性用户退出例程

- 表 29. 高级代理属性： IBM MQ 客户端压缩
- z/OS 表 30. 高级代理属性： z/OS -特定
- 表 31. 高级代理属性：其他属性

表 *66:* 高级代理程序属性: 代理程序大小

属性名	描述	缺省值
agentCheckpointInterval	为了恢复目的而采用检查点的间隔（以完整数据帧为单位）。 这是高级属性，对于大多数 Managed File Transfer 配置而言，无需修改其值。 如果存在导致传输进入恢复状态的问题，那么传输只可以恢复到检查点边界。 因此，该值越大（具有较大的 agentChunkSize、agentWindowSize 和 agentFrameSize 值），代理恢复传输所需的时间就越长。 对于传输很少进入恢复状态的可靠 Managed File Transfer 网络，增加该值可能很有益，这样可提高整体性能。	1
agentChunkSize	针对文件数据传输的每个传输块的大小。 因此，指示在源和目标代理之间传输的最大 IBM MQ 消息大小。 这是高级属性，对于大多数 Managed File Transfer 配置而言，无需修改其值。 该值是通过源代理和目标代理协商而定，使用两个值中较大一个。 如果要更改此属性的值，请在源代理和目标代理处更改该值。 agentChunkSize 是一个整数值。 例如：agentChunkSize=10240 将块大小设置为 10 KB。	262144 字节（等于 256KB）
agentFrameSize	传输帧的窗口数。 这是高级属性，对于大多数 Managed File Transfer 配置而言，无需修改其值。 对于具有较长等待时间的网络，增加此值可以提高整体性能，因为这会使代理同时具有更多活动消息块。 该属性的值乘以 agentWindowSize 再乘以 agentChunkSize，指示每个传输的代理内存使用量上限。 例如，针对每个传输，262144 字节块 x 10 x 5 = 12.5MB。 注：如果单个传输中传输的文件大小小于 12.5MB，那么增加该属性值对传输性能无影响。	5
agentWindowSize	每个窗口的块数。 这是高级属性，对于大多数 Managed File Transfer 配置而言，无需修改其值。 对于具有较长等待时间的网络，增加此值可以提高整体性能。 因为这会使代理同时具有更多活动消息块，并会降低将确认消息发送回源代理的频率。 该属性值乘以 agentFrameSize,），再乘以 agentChunkSize,表示代理每次传输的内存消耗上限，也表示目标代理数据队列中 "IBM MQ 消息数据的上限。 例如，针对每个传输，262144 字节块 x 10 x 5 = 上限 12.5MB。 注：如果单个传输中传输的文件大小小于 12.5MB，那么增加该属性值对传输性能无影响。	10

表 *67:* 高级代理程序属性: 代码页

属性名	描述	缺省值
agentCcsid	代理连接其代理队列管理器所使用的代码页。 如果为 agentCcsid 指定值，那么还必须为 agentCcsidName 指定值。 有关如何查看 JVM 的已知代码页的信息，请参阅 **fteCreateBridgeAgent** 命令中的 -hsc 参数 。	1208
agentCcsidName	agentCcsid 的 Java 表示。 如果为 agentCcsidName 指定值，那么还必须为 agentCcsid 指定值。	UTF8

表 *68:* 高级代理程序属性: 命令

属性名	描述	缺省值
maxCommandHandlerThreads	控制用于传输命令消息初始解析和处理的线程数。 当线程处于活动状态时，需要一个到队列管理器的连接，但在空闲时，线程会释放连接。	5
maxCommandOutput	用于命令输出存储的最大字节数。 该属性适用于为受管调用指定的命令以及为受管传输指定的 preSource、postSource、preDestination 和 postDestination 命令。 这会限制写入 SYSTEM.FTE 主题上传输日志的命令输出的长度。	10240
maxCommandRetries	代理允许重试命令的最大次数。 该属性适用于为受管调用指定的命令以及为受管传输指定的 preSource、postSource、preDestination 和 postDestination 命令。	9
maxCommandWait	代理允许重试之间的最长等待时间（秒）。 该属性适用于为受管调用指定的命令以及为受管传输指定的 preSource、postSource、preDestination 和 postDestination 命令。	60

表 68: 高级代理程序属性: 命令 (继续)

属性名	描述	缺省值
immediateShutdownTimeout	对于代理的立即关闭，您可以使用该属性来指定在强制关闭之前，代理等待完成传输的最长时间（秒）。 **注:** 请勿将该属性的值更改为小于缺省值（10 秒）。 代理的立即关闭需要足够的时间来终止任何外部进程。 如果该属性的值太低，那么进程可能仍保持运行。 如果为该属性指定值 0，那么代理会等待所有未完成的传输停止。 如果为该属性指定的值无效，那么将使用缺省值。	10

表 69: 高级代理程序属性: 连接

属性名	描述	缺省值
javaLibraryPath	当以绑定方式连接到队列管理器时， Managed File Transfer 必须具有对 IBM MQ Java 绑定库的访问权。 缺省情况下，Managed File Transfer 在 IBM MQ 定义的缺省位置中查找绑定库。 如果绑定库位于其他位置，那么请使用该属性指定绑定库的位置。	None

表 70: 高级代理程序属性: *Connect:Direct bridge*

属性名	描述	缺省值
cdNode	当您要使用 Connect:Direct 网桥时所需的属性。 用于将消息从 Connect:Direct 网桥代理传输到目标 Connect:Direct 节点的 Connect:Direct 节点的名称。 该节点属于 Connect:Direct 网桥，不是作为传输源或目标的远程节点。 有关更多信息，请参阅 Connect:Direct 网桥。	没有缺省值
cdNodeHost	用于将文件从 Connect:Direct 网桥代理传输到目标节点（Connect:Direct 网桥节点）的 Connect:Direct 节点的主机名或 IP 地址。 在大多数情况下，Connect:Direct 网桥节点与 Connect:Direct 网桥代理位于同一个系统上。 在这些情况下，该属性的缺省值（本地系统的 IP 地址）正确。 如果系统具有多个 IP 地址，或您的 Connect:Direct 网桥节点不在 Connect:Direct 网桥代理所在的系统上，并且它们的系统共享同一个文件系统，请使用该属性指定 Connect:Direct 网桥节点的正确主机名。 如果您未设置 cdNode 属性，那么会忽略该属性。	本地系统的主机名或 IP 地址
cdNodePort	客户机应用程序用于与其进行通信的 Connect:Direct 网桥节点的端口号。 在 Connect:Direct 产品文档中，该端口称为 API 端口。 如果您未设置 cdNode 属性，那么会忽略该属性。	1363
cimpDir	将文件传输到目标 Connect:Direct 节点之前在运行 Connect:Direct 网桥代理的系统上临时存储文件的位置。 该属性指定了临时存储文件的目录的完整路径。 例如，如果 cdTmpDir 设置为 `/tmp`，那么会将文件临时放置在 `/tmp` 目录中。 Connect:Direct 网桥代理和 Connect:Direct 网桥节点必须能够使用相同路径名访问该参数指定的目录。 请在计划安装 Connect:Direct 网桥时考虑这一点。 如果可能，请在作为 Connect:Direct 网桥的组成部分的 Connect:Direct 节点所在的系统上创建该代理。 如果代理和节点不在同一个系统上，那么该目录必须在共享文件系统上，并且可从两个系统使用相同路径名进行访问。 有关受支持配置的更多信息，请参阅 Connect:Direct 网桥。 如果您未设置 cdNode 属性，那么会忽略该属性。 此属性的值可以包含环境变量。 请参阅 第 145 页的『在 MFT 属性中使用环境变量』 以获取更多信息。	*value_of_java.io.tmpdir* `/cdbridge-`*agentName*
cdTrace	代理是否跟踪在 Connect:Direct 网桥代理与其 Connect:Direct 节点之间发送的数据。 该属性的值可以为 `true` 或 `false`。	false
cdMaxConnectionRetries	传输失败之前，针对尚未进行成功连接的文件传输的最大 Connect:Direct 连接尝试次数。	-1（无数次尝试）
cdMaxPartialWorkConnectionRetries	传输失败之前，针对先前连接尝试已成功并且传输工作已完成的文件传输的最大 Connect:Direct 连接尝试次数。	-1（无数次尝试）
cdMaxWaitForProcessEndStats	Connect:Direct 进程终止之后，断定文件传输已失败之前，等待进程完成信息在 Connect:Direct 节点统计信息内变为可用的最长时间（毫秒）。 通常，信息会立即可用，但在某些失败情况下，不会发布信息。 在这些情况下，等待由该属性指定的时间段后文件传输失败。	60000
cdAppName	Connect:Direct 网桥代理用于连接到属于网桥的 Connect:Direct 节点的应用程序名称。	Managed File Transfer *current version*，其中 *current version* 为产品的版本号。

表 70: 高级代理程序属性: *Connect:Direct bridge* (继续)		
属性名	**描述**	**缺省值**
cdNodeLocalPortRange	用于 Connect:Direct 网桥代理和属于网桥的 Connect:Direct 节点之间套接字连接的本地端口范围。 该值的格式为以逗号分隔的值列表或范围列表。 缺省情况下，操作系统选择本地端口号。	None
cdNodeProtocol	Connect:Direct 网桥代理用于连接到属于网桥的 Connect:Direct 节点的协议。 有效值包括: • TCPIP • SSL • TLS	TCPIP
cdNodeKeystore	Connect:Direct 网桥代理和属于该网桥的 Connect:Direct 节点之间安全通信所使用的密钥库的路径。 如果未将 cdNodeProtocol 属性设置为 SSL 或 TLS，那么会忽略该属性。 此属性的值可以包含环境变量。 请参阅 第 145 页的『在 MFT 属性中使用环境变量』 以获取更多信息。	None
cdNodeKeystoreType	cdNodeKeystore 属性指定的密钥库的文件格式。 以下值有效：jks 和 pkcs12。 如果未将 cdNodeProtocol 属性设置为 SSL 或 TLS，那么会忽略该属性。	jks
cdNodeKeystoreCredentialsFile	包含 cdNodeKeystore 凭证的文件的路径。 此属性的值可以包含环境变量。 请参阅 第 145 页的『在 MFT 属性中使用环境变量』 以获取更多信息。 ALW 有关创建凭证文件的详细信息，请参阅 MFT 和 IBM MQ 连接认证 z/OS 有关创建凭证文件的详细信息，请参阅 在 z/OS 上配置 MQMFTCredentials.xml。	此属性的缺省值为: z/OS Linux AIX $HOME/MQMFTCredentials.xml Windows %USERPROFILE%/MQMFTCredentials.xml
cdNodeTruststore	Connect:Direct 网桥代理和属于该网桥的 Connect:Direct 节点之间安全通信所使用的信任库的路径。 如果未将 cdNodeProtocol 属性设置为 SSL 或 TLS，那么会忽略该属性。 此属性的值可以包含环境变量。 请参阅 第 145 页的『在 MFT 属性中使用环境变量』 以获取更多信息。	None
cdNodeTruststoreType	cdNodeTruststore 属性指定的信任库的文件格式。 以下值有效：jks 和 pkcs12。 如果未将 cdNodeProtocol 属性设置为 SSL 或 TLS，那么会忽略该属性。	jks
cdNodeTruststoreCredentialsFile	包含 cdNodeTruststore 凭证的文件的路径。 此属性的值可以包含环境变量。 请参阅 第 145 页的『在 MFT 属性中使用环境变量』 以获取更多信息。 ALW 有关创建凭证文件的详细信息，请参阅 MFT 和 IBM MQ 连接认证 z/OS 有关创建凭证文件的详细信息，请参阅 在 z/OS 上配置 MQMFTCredentials.xml。	此属性的缺省值为: z/OS Linux AIX $HOME/MQMFTCredentials.xml Windows %USERPROFILE%/MQMFTCredentials.xml
logCDProcess	在 `output0.log` 文件的代理程序事件日志中记录的 Connect:Direct 进程日志记录级别。 该属性的值可以为 None 或 `Failures` 或 `All`。	None

表 71: 高级代理程序属性: 文件到消息，消息到文件代理程序		
属性名	**描述**	**缺省值**
deleteTmpFileAfterRenameFailure	将该属性的值设置为 false 可确保在重命名操作失败的情况下不会从目标中删除临时文件。 在此情况下，已传输的数据会作为临时 (.part) 文件保留在目标中。 稍后您可以手动重命名此文件。 缺省情况下，该属性的值为 true。 该属性同时应用于“消息到文件”传输和“文件到文件”传输。	true
enableQueueInputOutput	缺省情况下，在传输过程中，代理不能读取源队列中的数据或将数据写入目标队列。 将该值设置为 true，可使代理执行文件到消息和“消息到文件”传输。 该属性的值可以为 `true` 或 `false`。	false

表 71: 高级代理程序属性: 文件到消息，消息到文件代理程序 (继续)

属性名	描述	缺省值
enableSystemQueueInputOutput	指定代理是否可以读取或写入 IBM MQ 系统队列。 系统队列以限定词 SYSTEM 为前缀。 **注:** IBM MQ、Managed File Transfer 和其他应用程序使用系统队列来传输重要信息。 更改该属性，可使代理访问这些队列。 如果启用该属性，那么请使用用户沙箱来限制代理可以访问的队列。	false
enableClusterQueueInputOutput	指定代理是否可以读取或写入 IBM MQ 集群队列。 **注:** 除 enableQueueInputOutput 属性外，还必须指定 enableClusterQueueInputOutput 代理属性。	false
maxDelimiterMatchLength	可以通过用于在“文件到消息”传输过程中将文本文件拆分为多条消息的 Java 正则表达式进行匹配的最大字符数。	5
maxInputOutputMessageLength	由代理从源队列读取或写入目标队列的消息的最大长度（字节）。 传输中源代理的 maxInputOutputMessageLength 属性确定可以从源队列消息读取的字节数。 传输中目标代理的 maxInputOutputMessageLength 属性确定可写入到目标队列消息的字节数。 如果消息的长度超出该属性的值，那么传输失败并产生错误。 该属性不会影响 Managed File Transfer 内部队列。 有关更改此属性的信息，请参阅 "与报文大小相关的 "MQ 属性和 "MFT 属性设置指南。	1048576
monitorGroupRetryLimit	队列中仍存在消息组时，监视器再次触发“消息到文件”传输的最大次数。 触发“消息到文件”传输的次数由组中第一条消息的 MQMD 回退计数确定。 如果重新启动代理，那么即使触发传输的次数已超出 monitorGroupRetryLimit 的值，监视器仍会再次触发传输。 如果该行为导致触发传输的次数超出 monitorGroupRetryLimit 的值，那么代理会向其事件日志中写入一个错误。 如果对该属性指定值 -1，那么监视器会不限制次数地再次触发传输，直至不满足触发条件。	10

表 72: 高级代理程序属性: 常规

属性名	描述	缺省值
agentStatusPublishRateLimit	由于文件传输状态更改，代理重新发布其状态的最大速率（秒）。 如果将该属性设置为过小的值，那么 IBM MQ 网络的性能可能会受到负面影响。	30
agentStatusPublishRateMin	代理发布其状态的最小速率（秒）。 此值必须大于或等于 **agentStatusPublishRateLimit** 属性的值。	300
enableMemoryAllocationChecking	该属性的值可以为 `true` 或 `false`。 它确定 Managed File Transfer Agent 是否检查是否有足够的内存可用于在接受传输之前运行传输。 源和目标代理上都会执行该检查。 如果没有足够的内存，将拒绝此传输。 在计算传输所需的内存时，将使用传输所需的最大内存。 所以，该值可能大于传输使用的实际内存。 因此，如果 **enableMemoryAllocationChecking** 属性设置为 true ，那么可以运行的并发传输数可能会减少。 建议仅在遇到因内存不足错误而导致 Managed File Transfer 失败的问题时才将该属性设置为 true。 可能使用大量内存的传输包括消息大小较大的“文件到消息”传输和“消息到文件”传输。	false
enableDetailedReplyMessages	该属性的值可以为 `true` 或 `false`。 将此属性设置为 `true` 将使受管传输请求应答能够包含有关所传输文件的详细信息。 详细信息和格式与进度消息中发布到传输日志的相同，即 <transferSet> 元素。 有关更多信息，请参阅文件传输日志消息格式。 仅当受管传输请求指定详细回复信息为必需时，才会包含详细回复信息。 要指定这一要求，请在发送给源代理的 managedTransfer XML 请求信息的 <reply> 元素中设置 **detailed** 属性。 有关更多信息，请参阅文件传输请求消息格式。 可以为每个传输请求生成多条回复消息。 该数目等于传输的传输日志进度消息数目加 1（其中，第一条回复消息为简单 ACK 回复）。 除 ACK 回复消息外，所有消息中都包含详细信息，但总体传输结果仅包含在最后一条详细回复消息中。	true

表 72: 高级代理程序属性: 常规 (继续)

属性名	描述	缺省值
enableUserMetadataOptions	该属性的值可以为 `true` 或 `false`。 它确定是否可以将已知密钥用于新传输请求中的用户定义元数据，以提供更多传输选项。 这些已知键始终以前缀 `com.ibm.wmqfte.` 开头。 因此，当 **enableUserMetadataOptions** 属性设置为 `true` 时，用户定义的使用不支持使用此前缀的键。 当 **enableUserMetadataOptions** 属性设置为 `true` 时，当前支持的键如下所示: **com.ibm.wmqfte.insertRecordLineSeparator** 用于文本传输。 当此键设置为 `true` 时，指定在读取面向记录的文件(例如 z/OS 数据集) 时，将在记录之间插入行分隔符。 当此键设置为 `false` 时，指定在读取面向记录的文件时，不会在记录之间插入行分隔符。 **com.ibm.wmqfte.newRecordOnLineSeparator** 用于文本传输。 当此键设置为 `true` 时，指定当写入面向记录的文件(例如 z/OS 数据集) 时，行分隔符指示新记录并且不作为数据的一部分写入。 当此键设置为 `false` 时，指定在写入面向记录的文件时，将像对待任何其他字符一样对待行分隔符 (即，无记录分隔符)。 **com.ibm.wmqfte.convertLineSeparators** 用于文本传输。 指定是否将行分隔符序列 CRLF 和 LF 转换为目标所需的行分隔符序列。 此转换目前只在以下情况下生效: • 如果用户定义的元数据键 **com.ibm.wmqfte.newRecordOnLineSeparator** 设置为 `false`，并且传输到面向记录的文件。 • 如果用户定义的元数据键 **com.ibm.wmqfte.com.ibm.wmqfte.insertRecordLineSeparator** 设置为 `false`，并且传输来自面向记录的文件。 另请参阅 **fteCreateTransfer**: 启动新的文件传输。	false
failTransferOnFirstFailure	该属性的值可以为 `true` 或 `false`。 它允许将代理配置为在受管传输中的传输项失败时立即使该受管传输失败。 要启用此功能，必须对源代理和目标代理应用 APAR IT03450，并且必须在源代理的 `agent.properties` 文件中将 **failTransferOnFirstFailure** 属性设置为 `true`。 在目标代理上将此属性设置为 `true` 是可选的。 当 **failTransferOnFirstFailure** 属性设置为 `true` 时，代理将正常开始处理受管传输请求。 但是，一个传输项失败，就会将受管传输标记为失败，不会处理更多传输项。 将采用以下方式处理在受管传输失败前成功处理的传输项: • 采用这些传输项的源处置。 例如，如果已将传输项的源处置设置为 `delete`，那么将删除源文件。 • 编写的目标文件仍在目标文件系统上，不会将其删除。 如果 **failTransferOnFirstFailure** 属性未设置为 `true`，并且受管文件传输包含多个文件，并且其中一个文件无法传输，例如，由于目标文件已存在并且覆盖属性设置为 `error`，那么源代理将继续并尝试传输请求中的任何剩余文件。	false
itemsPerProgressMessage	代理发布其下一条进度日志消息之前传输的文件数。 此属性控制在传输期间将进度日志消息发布到协调队列管理器的速率。 该属性可以设置的最大值为 1000。 **注:** 进度消息包含有关发布上条进度消息之后传输的每个文件的信息。 增加此值将增加进度消息的大小，有可能会影响性能。	50
maxInlineFileSize	对于单个文件到文件或“文件到消息”传输，可以自动包含在初始传输请求消息中的最大文件大小（字节）。 您可以使用该属性提高传输速度，但如果将文件大小设置为过大的值，这可能会降低性能。 该属性的建议初始大小是 100KB，但建议彻底测试不同值，直到找到最适合系统的文件大小。 缺省情况下，将关闭此功能，或者通过将 **maxInlineFileSize** 属性设置为 `0` 来关闭此功能。	0

表 73: 高级代理程序属性: 高可用性

属性名	描述	缺省值
highlyAvailable	此属性在代理程序启动期间读取，如果设置为值 `true`，那么将以高可用性方式启动代理程序。 如果未指定此属性，或者将值设置为 false，那么代理程序将作为非高可用性代理程序启动。	false

表 73: 高级代理程序属性: 高可用性 (继续)		
属性名	**描述**	**缺省值**
standbyPollInterval	备用实例使用此属性来尝试按指定的时间间隔打开共享队列。 从 IBM MQ 9.3.0 开始，所有实例也使用此属性来确定实例在从其代理队列管理器断开连接时，在两次重新连接尝试之间等待的时间长度。 将重复这些尝试，直到实例重新连接到其代理队列管理器，打开 SYSTEM.FTE.HA.<agent name> 队列 (如果它已将自身注册为备用实例) 或通过 **fteStopAgent** 命令停止为止。	5 秒
standbyStatusDiscardTime	此属性设置活动实例等待来自备用实例的状态发布的持续时间。 如果未从备用实例接收到任何发布，那么即使在此既成时间之后，活动实例也会从其备用实例列表中除去备用实例信息。 缺省值是 **standbyStatusPublishInterval** 属性的两倍，以便活动实例在从其列表中除去备用实例之前等待更长时间。	600 秒
standbyStatusExpiry	此属性设置将备用状态消息放入代理的命令队列的到期时间。 如果代理程序的活动实例未处理此消息，那么此消息将到期。	30 秒
standbyStatusPublishInterval	此属性用于设置备用实例发布其状态的频率。	300 秒

表 74: 高级代理程序属性: 输入/输出		
属性名	**描述**	**缺省值**
doNotUseTempOutputFile	缺省情况下，代理将写入目标上的临时文件中，并在文件传输完成后，将该临时文件重命名为所需的文件名。 将该值设置为 true 会使代理对最终的目标文件进行直接写入。 z/OS 在 z/OS 系统上，该行为不会应用于连续数据集，但会应用于 PDS 数据集成员。 该属性的用于传输的值由目标代理定义。	false
enableMandatoryLocking	访问常规文件时，Managed File Transfer 对读取操作采取共享锁定，对写入操作采取互斥锁定。 Windows 在 Windows 上，只是建议文件锁定。 将该属性设置为 true 时，Managed File Transfer 会强制文件锁定。 在 Windows 上，这意味着如果另一个应用程序打开一个文件，那么不会触发监视该文件，直至文件关闭为止。 Managed File Transfer 传输使文件失败。 UNIX 在 UNIX 类型的平台上，进程之间采用文件锁定。 对于 UNIX 类型平台，设置此属性没有任何影响。 该属性仅适用于普通 Managed File Transfer 代理。 Managed File Transfer 不支持文件服务器上的文件锁定机制。 因此，该属性对于协议网桥代理无效，因为协议网桥代理在传输文件时不会锁定文件服务器上的文件。 该属性的值可以为 `true` 或 `false`。	false
ioIdleThreadTimeout	线程关闭之前，文件系统输入/输出线程保持空闲的时间（毫秒）。 z/OS 从 IBM MQ 9.3.0 开始，此属性不适用于在 IBM MQ for z/OS 上运行的代理程序。	10000
ioQueueDepth	排队的最大输入/输出请求数。	10
ioThreadPoolSize	可用的最大文件系统输入/输出线程数。 通常，每个传输都使用它自己的文件系统输入/输出线程，但是如果并行传输数超过此限制，那么会在传输之间共享文件系统输入/输出线程。 如果您认为您可能会经常具有比 ioThreadPoolSize 值更多的并行传输，那么您通过增加该值会看到改进，以便每个传输都有它自己的文件系统输入/输出线程。	10
textReplacementCharacterSequence	对于文本方式传输，如果不能将任何数据字节从源代码页转换到目标代码页，那么缺省行为为文件传输失败。 通过插入指定的字符值，将该属性设置为允许传输成功完成。 该属性值是单字符。 通常，问号 (?) 用于表示任何不可映射的字符。 例如，使用以下格式: textReplacementCharacterSequence=?，其中问号 (?) 是替换字符。 不能使用空格字符作为替换字符。	None

表 75: 高级代理属性: 传输日志		
属性名	**描述**	**缺省值**
logTransfer 传输事件采用 JSON 格式，并写入代理日志目录下的 `transferlog`*n*`.json` 文件，其中 *n* 的默认值为 0。 有关生成日志信息的示例，请参阅 第 181 页的『由 LogTransfer 函数生成的输出』。	打开或关闭传输日志记录。 可能的值为： **info** 启用传输的简要日志信息摘要。 这是缺省值。 **moderate** 启用传输的中间级别日志信息。 **verbose** 启用传输的详细日志信息。 **关** 关闭传输日志记录	info
logTransferFileSize	定义传输日志文件的最大大小 (以兆字节为单位)	20
logTransferFiles	定义在废弃最旧的文件之前保留的最大传输文件数。	5

表 76: 高级代理程序属性: 多通道支持		
属性名	**描述**	**缺省值**
agentMultipleChannelsEnabled	将此属性设置为 `true` 将使 Managed File Transfer Agent 能够通过多个 IBM MQ 通道发送传输数据消息。 在某些情况下，设置该属性可提高性能。 但是，只有当存在显著的性能优势时，才应启用多通道支持。 在多个通道之间只发送置于 SYSTEM.FTE.DATA.*destinationAgentName* 队列的消息。 对所有其他消息的行为保持不变。 将该属性设置为 `true` 时，您还必须在以下某个主题中完成 IBM MQ 配置步骤才能启用多通道支持： • 在集群中为多个通道配置 MFT 代理 • 为多个通道配置 MFT 代理：非集群 此外，您还必须完成 Managed File Transfer 代理程序所需的标准 IBM MQ 配置步骤，这些步骤在 配置 MFT 以首次使用中进行了详细说明。 该属性的值可以为 `true` 或 `false`。	false
agentMessageBatchSize	配置多通道后，源代理以循环法为每个通道之间的传输发送数据消息。 该属性控制每次沿每个通道发送的消息数。	5

表 77: 高级代理属性: 多实例队列管理器		
属性名	**描述**	**缺省值**
agentQMgrStandby	由 agentQMgr 定义的多实例代理队列管理器的备用实例在建立客户机连接时所使用的主机名和端口号，采用 IBM MQ CONNAME 格式。 例如，*host_name*(*port_number*) 当代理程序检测到连接中断错误 (例如 MQRC 2009) 时，它会尝试连接到备用队列管理器。 代理连接到备用队列管理器后，代理将保持连接状态，直到备用队列管理器变为不可用为止。	没有缺省值

表 78: 高级代理程序属性: 进程控制器		
属性名	**描述**	**缺省值**
agentQMgrRetryInterval	代理的进程控制器对队列管理器的可用性进行检查的时间间隔（秒）。	30
maxRestartCount	在 maxRestartInterval 属性值所指定的时间间隔内可发生的最大重新启动次数。 当超出该值时，代理的进程控制器会停止重新启动代理，而改为执行基于 maxRestartDelay 属性值的操作。	4
maxRestartInterval	代理的进程控制器度量代理重新启动的时间间隔（秒）。 如果该时间间隔内的重新启动次数超过 maxRestartCount 属性的值，那么代理的进程控制器会停止重新启动代理。 代理的进程控制器会改为执行基于 maxRestartDelay 属性值的操作。	120
maxRestartDelay	确定代理的进程控制器在代理重新启动的速率超过 maxRestartCount 和 maxRestartInterval 属性值时的行为。 如果指定小于或等于零的值，那么代理的进程控制器会停止。 如果指定大于零的值，那么这即是重置代理的进程控制器持有的重新启动历史记录信息并重新启动代理之前等待的秒数。	-1

表 79: 高级代理程序属性: 协议网桥

属性名	描述	缺省值
protocolBridgeCredentialConfiguration	该属性的值作为字符串传入由 protocolBridgeCredentialExitClasses 指定的出口类的 initialize() 方法。	空值
protocolBridgeCredentialExitClasses	指定实现了协议网桥凭证用户出口例程的类的逗号分隔列表。 有关更多信息，请参阅使用出口类映射文件服务器的凭证。	没有缺省值。
protocolBridgeDataTimeout	协议网桥代理等待建立通往 FTP 服务器的数据连接或通过已经建立的连接接收来自 FTP 服务器的数据的超时时间（以毫秒为单位）。 如果将此属性值设置为 0，协议网桥代理将会无限期等待。 如果到达超时时间，则协议网桥代理将关闭通往 FTP 服务器的任何现有数据连接，并在恢复当前传输之前尝试建立新数据连接。 如果建立新数据连接的尝试失败，那么当前传输也会失败。	0
protocolBridgeLogoutBeforeDisconnect	指定协议网桥代理在关闭 FTP 会话并断开连接之前是否会使用用户从文件服务器注销。 如果将此属性设置为 `true`，协议网桥代理将向文件服务器发出 FTP **`QUIT`** 命令。	false
protocolBridgePropertiesConfiguration	作为某个网桥属性传递到由 protocolBridgeServerPropertiesExitClasses 属性指定的出口类的 initialize() 方法。	没有缺省值
protocolBridgePropertiesExitClasses	指定实现了协议网桥服务器属性用户出口例程的类的逗号分隔列表。 有关更多信息，请参阅 ProtocolBridgePropertiesExit2：查找协议文件服务器属性。	没有缺省值

表 80: 高级代理程序属性: 协议网桥代理日志记录

属性名	描述	缺省值
agentLog	用于在协议网桥代理和 FTP/SFTP/FTPS 文件服务器之间启用或禁用 FTP 命令和响应的日志记录的键值对组件及操作。 例如: **agentLog=on** 打开所有组件的日志记录 **agentLog=off** 关闭所有组件的日志记录 **agentLog=ftp=on, sftp=on, ftps=off** 打开 FTP 和 SFTP 的日志记录，并关闭 FTPS 的日志记录	没有缺省值
agentLogFileSize	定义捕获文件的最大大小（兆字节）。 与常规跟踪缺省文件大小的缺省值相同。	20
agentLogFiles	定义在废弃最旧的文件之前保留的最大捕获文件数。 • **agentLogFiles** 代理程序属性的缺省值已从 10 更改为 5。 这意味着从 IBM MQ 9.3.0 开始，如果设置了缺省值，那么最多可以有 5 个协议网桥代理事件日志文件，从 `agentevent0.log` 到 `agentevent4.log`。 但是，如果需要，您可以更改此值。 • 如果代理程序是从 IBM MQ 9.3.0 之前的版本迁移的，那么应手动删除 `agentevent5.log` 到 `agentevent9.log` 文件 (如果存在)。 • 但是，每个日志文件的大小仍为 20 MB。	从 IBM MQ 9.3.0 开始，缺省值为 5。

agentLogFilter	缺省情况下，捕获与代理连接到的所有 FTP 服务器的通信。
	例如:
	• 对主机/IP 地址进行过滤
	`host=ftpprod.ibm.com, ftp2.ibm.com`
	`host=9.182.*`
	• 根据元数据进行过滤
	`metadata="outbound files to xyz corp"`
		*

表 81: 高级代理程序属性: 队列

属性名	描述	缺省值
dynamicQueuePrefix	此属性定义创建临时动态队列时要使用的前缀。	WMQFTE.*
modelQueueName	此属性定义创建临时动态队列时要使用的模块队列的名称。	SYSTEM.DEFAULT.MODEL.QUEUE
publicationMDUser	与协议队列管理器发布的已发送消息关联的 MQMD 用户标识。 如果您未设置该属性，那么根据用于设置 MQMD 用户标识的 IBM MQ 规则设置 MQMD 用户标识。	没有缺省值

表 *82:* 高级代理程序属性: 资源监视		
属性名	**描述**	**缺省值**
monitorFilepathPlatformSeparator	指定是否在 *$FILEPATH* 变量中使用特定于平台的路径分隔符。 `true` 值表示使用特定于平台的路径分隔符。 值为 `false` 表示在所有平台上都使用 UNIX 样式的正斜杠 (/) 路径分隔符。	true
monitorMaxResourcesInPoll	指定在每个轮询时间间隔内触发的最大资源监视数。 例如，如果将监视模式指定为 `*.txt`，将轮询时间间隔指定为 10 秒并将 monitorMaxResourcesInPoll 属性设置为 10，那么 monitorMaxResourcesInPoll 属性会限制代理每个轮询时间间隔最多触发 10 个匹配项。 在以后的轮询时间间隔内会触发 10 个限制以上的匹配资源。 另外，您可以将 monitorMaxResourcesInPoll 属性与 **`fteCreateMonitor`** 命令的匹配 **-bs** 参数（例如）结合使用，以限制每个轮询时间间隔只触发一个传输。 小于或等于零的值表示在轮询时间间隔内触发的监视资源数不受限制。	-1
monitorReportTriggerFail	指定是否将在监视器中检测到的环境和配置中的失败情况作为日志消息报告给 SYSTEM.FTE 主题。 `true` 值记录消息。 `false` 值不记录消息。	true
monitorReportTriggerNotSatisfied	指定不满足条件的触发器是否向包含详细信息的 SYSTEM.FTE 主题发送日志消息。 `true` 值记录消息。 `false` 值不记录消息。	false
monitorReportTriggerSatisfied	指定满足条件的触发器是否向包含详细信息的 SYSTEM.FTE 主题发送日志消息。 `true` 值记录消息。 `false` 值不记录消息。	false
monitorSilenceOnTriggerFailure	不再报告失败之前，资源监视器触发器的连续失败数。	5
monitorStopOnInternalFailure	资源监视器将其状态更改为停止之前，资源监视器的连续内部 FFDC 条件数。	10

表 *83:* 高级代理程序属性: 根目录		
属性名	**描述**	**缺省值**
commandPath	指定可调用命令的路径集，可以通过使用一下方法之一： • 代理程序 Ant fte: call Ant task， fte: filecopy 或 fte: filemove 任务 • 在传递给代理的 XML 消息中，通过使用某个受支持的 Managed File Transfer Agent 命令 XML 模式（例如，managedCall 或 managedTransfer）。 有关 commandPath 属性值的有效语法的信息，请参阅 commandPath MFT property。 **要点:** 设置此属性时请非常小心，因为可以从能够向代理程序发送命令的远程客户机系统有效调用其中一个指定 **`commandPath`** 中的任何命令。 因此，缺省情况下，当您指定 **`commandPath`** 时: • 代理在启动时配置任何现有代理沙箱，以便将所有 **`commandPath`** 目录自动添加到已拒绝传输访问的目录列表中。 • 代理程序启动时，将更新任何现有用户沙箱，以便将所有 **`commandPath`** 目录 (及其子目录) 作为 `<exclude>` 元素添加到 `<read>` 和 `<write>` 元素。 • 如果未将代理程序配置为使用代理程序沙箱或用户沙箱，那么在代理程序启动时将创建新的代理程序沙箱，并将 **`commandPath`** 目录指定为已拒绝的目录。 此属性的值可以包含环境变量。 请参阅 第 145 页的『在 MFT 属性中使用环境变量』 以获取更多信息。 您可以将 **`addCommandPathToSandbox`** 属性设置为 false 以覆盖此缺省行为。 **要点:** 请注意，此覆盖实际上支持客户机将任何命令传输到代理系统并调用该命令，因此在使用时务必谨慎。	None - 不能调用任何命令
addCommandPathToSandbox	指定是否应将 **`commandPath`** 属性指定的目录 (及其所有子目录) 添加到: • 适用于现有代理沙箱的拒绝的目录。 • 对于已定义的任何用户沙箱，`<read>` 和 `<write>` 元素的 `<exclude>` 元素。 • 新的代理沙箱，如果没有为代理配置一个代理沙箱或者一个或多个用户沙箱。 有关 **`commandPath`** 属性值的有效语法的信息，请参阅 commandPath MFT property。	True

表 83: 高级代理程序属性: 根目录 (继续)

属性名	描述	缺省值
additionalWildcardSandboxChecking	指定是否对使用用户或代理沙箱配置的代理的通配符传输进行额外检查，以便限制代理可以将文件传输到以及从中传输文件的位置。 将该属性设置为 true 时，会启用额外检查。 如果传输请求尝试读取进行通配符文件匹配的已定义沙箱外部的位置，那么传输将失败。 如果一个传输请求中存在多个传输，并且其中一个请求因尝试读取沙箱外部的位置而失败，那么整个传输也将失败。 如果检查失败，那么会在错误消息中给出失败原因 (请参阅 通配符传输的其他检查)。 如果忽略此属性或者将其设置为 false，那么将不会对通配符传输进行额外检查。	None
sandboxRoot	指定使用沙箱时包含和排除的根路径集。 有关此功能的信息，请参阅 使用 MFT 代理程序沙箱 。 使用特定于平台的路径分隔符分隔路径。 路径以惊叹号 (!) 字符为前缀以表示从沙箱排除路径。 当您要在包含的根路径下排除子目录时，该功能很有用。 sandboxRoot 属性在协议网桥代理上不受支持。 您不能同时指定 sandboxRoot 属性和 userSandboxes 属性。 此属性的值可以包含环境变量。 请参阅 第 145 页的『在 MFT 属性中使用环境变量』 以获取更多信息。	None - 没有沙箱
transferRoot	为代理指定的相对路径的缺省根目录。 此属性的值可以包含环境变量。 请参阅 第 145 页的『在 MFT 属性中使用环境变量』 以获取更多信息。	启动代理进程的用户的主目录。
transferRootHLQ	为代理指定的非标准数据集的缺省 HLQ（用户标识）	启动代理进程的用户的用户名。
userSandboxes	根据请求传输的用户的 MQMD 用户名，限制可以传输文件的文件系统区域。 有关更多信息，请参阅 使用 MFT 用户沙箱。 userSandboxes 属性在协议网桥代理上不受支持。 您不能同时指定 sandboxRoot 属性和 userSandboxes 属性。	false

表 84: 高级代理属性: 调度程序属性

属性名	描述	缺省值
maxSchedulerRunDelay	代理等待检查已调度传输的最大时间间隔（分钟）。 指定一个正整数以启用该属性。 有关可能要使用此属性的原因的更多信息，请参阅 在调度的文件传输未运行或延迟时要执行的操作。 因为代理可能在调度的传输将要运行时从其命令队列读取命令，因此在启动调度的传输之前，可能存在额外的延迟。 在此情况下，调度程序将在该命令完成后立即运行。	-1

表 85: 高级代理程序属性: 安全性

属性名	描述	缺省值
agentCredentialsKeyFile	包含加密凭证时使用的凭证密钥的文件的名称。	没有缺省值的字符串属性。
agentQMgrAuthenticationCredentialsFile	包含连接到代理队列管理器时应使用的凭证的文件的路径。 此属性的值可以包含环境变量。 请参阅 第 145 页的『在 MFT 属性中使用环境变量』 以获取更多信息。 ALW 有关创建凭证文件的详细信息，请参阅 MFT 和 IBM MQ 连接认证 z/OS 有关创建认证凭证文件的详细信息，请参阅 在 z/OS 上配置 MQMFTCredentials.xml。	此属性的缺省值为: z/OS Linux AIX $HOME/MQMFTCredentials.xml Windows %USERPROFILE%/MQMFTCredentials.xml
authorityChecking	指定是否启用 限制 MFT 代理操作的用户权限 中描述的安全功能。 inquire 许可权是对所有代理权限队列的必需许可权。	false
logAuthorityChecks	output0.log 文件的代理事件日志中记录的权限检查记录的级别。 该属性的值可以为 None 或 Failures 或 All。	None
userIdForClientConnect	经客户机连接流至 IBM MQ 的用户标识。 如果指定了 *java*，那么 JVM 报告的用户名会作为 IBM MQ 连接请求的一部分流动。 该属性的值可以为 None 或 java。	None

表 86: 高级代理程序属性: *SSL/TLS*		
属性名	**描述**	**缺省值**
agentSslCipherSpec	指定在代理和代理队列管理器之间交换数据时，使用的协议、散列算法和加密算法，以及加密密钥中使用的位数。 agentSslCipherSpec 的值为 CipherSpec 名称。 该 CipherSpec 名称与代理队列管理器通道中使用的 CipherSpec 名称相同。 Java 的 "IBM MQ 类中的 SSL/TLS "CipherSpecs 和 "CipherSuites"和 "JMS 的 "IBM MQ 类中的 SSL/TLS "CipherSpecs 和 "CipherSuites"中包含有效的 "CipherSpec 名称列表。 agentSslCipherSpec 类似于 agentSslCipherSuite。 如果指定了 agentSslCipherSuite 和 agentSslCipherSpec，那么使用 agentSslCipherSpec 的值。	None
agentSslCipherSuite	指定代理程序和代理程序队列管理器交换数据方式的 SSL 方面。 agentSslCipherSuite 的值为 CipherSuite 名称。 CipherSuite 名称将映射到代理队列管理器通道中使用的 CipherSpec 名称。 有关更多信息，请参阅 CipherSuite 和 CipherSpec 名称映射。 agentSslCipherSuite 类似于 agentSslCipherSpec。 如果指定了 agentSslCipherSuite 和 agentSslCipherSpec，那么使用 agentSslCipherSpec 的值。	None
agentSslPeerName	指定必须与代理队列管理器提供的名称相匹配的专有名称框架。 专有名称用于检查连接时队列管理器提供的标识证书。	None
agentSslTrustStore	指定代理程序信任的证书的位置。 agentSslTrustStore 的值为文件路径。括号、逗号 (,) 和反斜线 (\) 是 MFT 命令中的特殊字符，必须用反斜线 (\) 字符转义。 Windows Windows 上的文件路径可以使用双反斜线 (\\) 作为分隔符，也可以使用单个正斜线 (/) 指定。 此属性的值可以包含环境变量。	None
agentSslKeyStore	指定代理程序专用密钥的位置。 agentSslKeyStore 的值为文件路径。括号、逗号 (,) 和反斜线 (\) 是 MFT 命令中的特殊字符，必须用反斜线 (\) 字符转义。 Windows Windows 上的文件路径可以使用双反斜线 (\\) 作为分隔符，也可以使用单个正斜线 (/) 指定。 仅当代理队列管理器需要客户机认证时才需要该属性。 此属性的值可以包含环境变量。	None
agentSslFipsRequired	指定要在代理级别启用 FIPS 支持。 该属性的值可以为 `true` 或 `false`。 更多信息，请参阅 FIPS 支持 MFT。	false
agentSslKeyStoreType	您要使用的 SSL 密钥库的类型。 支持 JKS 和 PKCS#12 密钥库。 该属性的值可以为 `jks` 或 `pkcs12`。	jks
agentSslKeyStoreCredentialsFile	包含用于访问代理程序的密钥库的凭证的文件的路径。 此属性的值可以包含环境变量。 请参阅 第 145 页的『在 MFT 属性中使用环境变量』 以获取更多信息。 ALW 有关创建凭证文件的详细信息，请参阅 MFT 和 IBM MQ 连接认证 z/OS 有关创建认证凭证文件的详细信息，请参阅 在 z/OS 上配置 MQMFTCredentials.xml。	此属性的缺省值为: z/OS Linux AIX `$HOME/MQMFTCredentials.xml` Windows `%USERPROFILE%/MQMFTCredentials.xml`
agentSslTrustStoreType	您要使用的 SSL 密钥库的类型。 支持 JKS 和 PKCS#12 密钥库。 该属性的值可以为 `jks` 或 `pkcs12`。	jks
agentSslTrustStoreCredentialsFile	包含用于访问代理程序的信任库的凭证的文件的路径。 此属性的值可以包含环境变量。 请参阅 第 145 页的『在 MFT 属性中使用环境变量』 以获取更多信息。 ALW 有关创建凭证文件的详细信息，请参阅 MFT 和 IBM MQ 连接认证 z/OS 有关创建认证凭证文件的详细信息，请参阅 在 z/OS 上配置 MQMFTCredentials.xml。	此属性的缺省值为: z/OS Linux AIX `$HOME/MQMFTCredentials.xml` Windows `%USERPROFILE%/MQMFTCredentials.xml`

表 87: 高级代理程序属性: 超时

属性名	描述	缺省值
maxTransferNegotiationTime	传输等待目标代理完成协商的最长时间（毫秒）。 如果在这期间未完成协商，那么传输会进入再同步状态，并允许其他传输（如果可用）运行。 在源或目标代理负载过重的情况下，可能是针对代理的缺省值太低以致无法足够快速地响应协商请求。 当源代理定义了大量资源监视器或其资源监视器正在监视包含大量文件的目录时，最有可能发生此情况。 但是，在向代理提交大量传输请求时，也可能会发生此情况。 在此类情况下，可能需要将该属性的值增加至 200,000 或更大值。	30 000
recoverableTransferRetryInterval	检测可恢复传输错误和尝试恢复传输之间等待的时间（毫秒）。	60 000
senderTransferRetryInterval	由于目标已运行最大数量的传输而重试拒绝的传输之前等待的时间（毫秒）。 最小值为 1000。	30 000
transferAckTimeout	发出重试之前，传输等待来自另一端的应答或数据的超时时间（毫秒）。 这是高级属性，对于大多数 Managed File Transfer 配置而言，无需修改其值。 只要收到完整的数据窗口，就向发送代理发送来自接收代理的应答。 对于带宽受到限制或不可靠的网络以及大的 agentWindowSize 和 agentChunkSize 设置，缺省值可能不够长。 这可能会导致代理之间不必要的数据重新传输。 因此，增加该值可能是有益的，可能会降低由于网络速度缓慢而使传输进入恢复方式的可能性。	60 000
transferAckTimeoutRetries	代理放弃并将传输移至恢复状态之前，在没有响应的情况下，传输的最大应答重试次数。	5
xmlConfigReloadInterval	运行时期间代理重新载入 XML 配置文件之间的时间间隔（秒）。 为防止代理在运行时期间重新载入 XML 配置文件，请将该属性设置为 -1。 该属性将影响以下 XML 配置文件: • ConnectDirectCredentials.xml • ConnectDirectNodeProperties.xml • ConnectDirectProcessDefinitions.xml • ProtocolBridgeCredentials.xml • ProtocolBridgeProperties.xml • UserSandboxes.xml	30

表 88: 高级代理程序属性: 跟踪和日志记录

属性名	描述	缺省值
javaCoreTriggerFile	代理监视的文件位置的完整路径。 如果文件存在于指定位置，那么代理程序启动将触发 Javacore。 启动代理程序后，如果在此位置更新文件，那么代理程序将再次触发 Javacore 文件。 单独的线程会每 30 秒轮询该文件以检查是否已创建或更新文件。 如果自上次轮询以来已创建或更新该文件，那么代理程序将在以下目录中生成 Javacore 文件: *MQ_DATA_PATH*/mqft/logs/*coordination_qmgr_name*/agents/*agent_name* 指定该属性后，代理会在启动时输出以下消息: `BFGAG0092I The <insert_0> file will be used to request JVM diagnostic information.` 此属性的值可以包含环境变量。 请参阅 第 145 页的『在 MFT 属性中使用环境变量』 以获取更多信息。	None

表 88: 高级代理程序属性: 跟踪和日志记录 (继续)

属性名	描述	缺省值

跟踪	启动代理程序时要使用的跟踪规范。 这是以逗号分隔的类和/或包，等于字符和跟踪级别的列表。
	例如，要从代理程序启动开始跟踪 `com.ibm.wmqfte.agent.Agent` 类和 `com.ibm.wmqfte.commandhandler` 包中的类，请将以下条目添加到 `agent.properties` 文件中:
	`trace=com.ibm.wmqfte.agent.Agent,com.ibm.wmqfte.commandhandler=all`
	您可以在冒号分隔的列表中指定多个跟踪规范。 例如
	`trace=com.ibm.wmqfte.agent.Agent=all:com.ibm.wmqfte.commandhandler=moderate`
	特殊跟踪规范 `=all` 用于跟踪代理程序以及用于处理与代理队列管理器的所有通信的 Java 消息队列接口 (JMQI)。 要启用此功能，请将以下条目添加到 `agent.properties` 文件:
	`trace==all`
	除非 "IBM 支持中另有说明，否则应这样使用跟踪规范 `"com.ibm.wmqfte=all`
	`trace=com.ibm.wmqfte=all`
		None

outputLogFiles	保存的 `output.log` 文件的总数。 该值适用于代理的进程控制器以及代理自身。	5
outputLogSize	将输出合并到下一个文件之前，每个 `output.log` 文件的最大大小 (MB)。 该值适用于代理的进程控制器以及代理自身。	1
outputLogEncoding	写入 `output.log` 文件时，代理使用的字符编码。	正在运行代理的平台的缺省字符编码。
traceFiles	保存的跟踪文件的总数。 该值适用于代理的进程控制器以及代理自身。	5
traceSize	将跟踪合并到下一个文件之前，每个跟踪文件的最大大小 (MB)。 该值适用于代理的进程控制器以及代理自身。	20
traceMaxBytes	对在跟踪文件中输出的消息数据数量的限制。	4096 字节
logTransferRecovery	如果将该属性的值设置为 true，每当传输进入恢复状态时，都会向 `output0.log` 文件中代理的事件日志报告诊断事件。	缺省值为 true。
logCapture	捕获提交给该代理的传输请求消息和代理发布到协调队列管理器的日志消息。 在调试传输问题时，捕获的这些消息可能会很有帮助。 捕获的信息存储在代理日志目录中名为 `captureN.log` 的文件中。 *n* 是一个数值。 包含数字 `0` 的文件保存最新捕获的消息。	false
logCaptureFileSize	定义捕获文件的最大大小（兆字节）。	10
logCaptureFiles	定义在废弃最旧的文件之前保留的最大捕获文件数。	10
logCaptureFilter	代理用于与消息的主题名称匹配的 Java 正则表达式。 仅捕获那些与正则表达式匹配的消息。	.* (match all)
resourceMonitorLog	用于开启或关闭日志记录的资源监视器和操作的键值对。 可能的值为: • info • moderate • verbose • 关 例如: • `resourceMonitorLog=MON1,MON2=info:MON3=off` 开启 `MON1` 和 `MON2` 的日志记录，并关闭 `MON3` 的日志记录。 • `resourceMonitorLog=info` 对所有资源监视器开启 `info` 级别日志记录。 资源监视器日志将写入名为 `resmonevent`*N*`.log` 的文件，其中 *N* 代表数字; 例如， `resmonevent0.log`。 **注意:** 代理的所有资源监视器将写入同一个日志文件中。 请参阅记录 MFT 资源监视器，以获取更多信息。	info

表 *88:* 高级代理程序属性: 跟踪和日志记录 (继续)

属性名	描述	缺省值
resourceMonitorLogFileSize	定义捕获文件的最大大小（兆字节）。	20
resourceMonitorLogFiles	定义在废弃最旧的文件之前保留的最大捕获文件数。 • **resourceMonitorLogFiles** 代理程序属性的缺省值已从 10 更改为 5。 这意味着，从 IBM MQ 9.3.0 开始，如果设置了缺省值，那么最多可以有 5 个资源监视器事件日志文件，从 `resmonevent0.log` 到 `resmonevent4.log`。 但是，如果需要，您可以更改此值。 • 如果代理程序是从 IBM MQ 9.3.0 之前的版本迁移的，那么应手动删除 `resmonevent5.log` 到 `resmonevent9.log` 文件 (如果存在)。 • 但是，每个日志文件的大小仍为 20 MB。	从 IBM MQ 9.3.0 开始，缺省值为 5。

表 *89:* 高级代理属性: 传输限制

属性名	描述	缺省值
maxDestinationTransfers	目标代理在任意时间点处理的最大并行传输数。 向代理提交的每个传输请求都会针对该总数计数，而与满足该请求的传输文件数无关。 这意味着传输一个文件的传输请求与传输 10 个文件的传输请求的计数方法相同。 如果目标代理达到 maxDestinationTransfers 属性指定的限制，那么此代理会将传输排队。 如果代理属性值 maxSourceTransfers + maxDestinationTransfers + maxQueuedTransfers 的总和 超出状态存储队列 (SYSTEM.FTE.STATE.*agent name*) 的 MAXDEPTH 设置值，那么不启动代理。	25（对于 Connect:Direct 之外的所有代理） 5（对于 Connect:Direct 网桥代理）
maxFilesForTransfer	单个受管传输允许的最大传输项数。 如果受管传输包含比 maxFilesForTransfer 的值更多的项，那么受管传输将失败并且不会处理任何传输项。 设置此属性可以防止您因传输请求错误而意外传输过多文件，例如，如果用户在 AIX and Linux 系统上意外指定传输根目录 / 。	5000
maxSourceTransfers	源代理在任何时间点处理的最大并行传输数。 向代理提交的每个传输请求都会针对该总数计数，而与满足该请求的传输文件数无关。 这意味着传输一个文件的传输请求与传输 10 个文件的传输请求的计数方法相同。 如果目标代理达到 maxSourceTransfers 属性指定的限制，那么源代理会将传输排队。 如果代理属性值 maxSourceTransfers + maxDestinationTransfers + maxQueuedTransfers 的总和 超出状态存储队列 (SYSTEM.FTE.STATE.*agent name*) 的 MAXDEPTH 设置值，那么不启动代理。	25（对于 Connect:Direct 网桥代理之外的所有代理） 5（对于 Connect:Direct 网桥代理）
maxQueuedTransfers	在代理拒绝新的传输请求之前，源代理可排队的暂挂传输的最大数目。 您可以设置该属性，以便不管满足还是超出 maxDestinationTransfers 和 maxSourceTransfers 的限制，都接受您此时提交的任何新传输请求，将其排队，并在稍后执行。 排队的传输请求的处理顺序取决于其优先级及排队的时间长短。 首先，会选择旧优先级和高优先级暂挂的传输。 队列中具有低优先级但排队时间较长的传输优先于较新且具有较高优先级的传输。 如果代理属性值 maxSourceTransfers + maxDestinationTransfers + maxQueuedTransfers 的总和 超出状态存储队列 (SYSTEM.FTE.STATE.*agent name*) 的 MAXDEPTH 设置值，那么不启动代理。	1000

表 *90:* 高级代理程序属性: 传输恢复超时

属性名	描述	缺省值
transferRecoveryTimeout	设置时间量（以秒为单位），在此期间，源代理会一直尝试恢复停止的文件传输。 未设置此属性时，代理的缺省行为是一直重试，直至成功恢复传输。 您可以为传输恢复超时属性设置以下值: **-1** 代理继续尝试恢复停止的传输，直至传输完成。 使用此选项相当于未设置属性时代理的缺省行为。 **0** 一旦进入恢复，代理将停止文件传输。 **>0** 在由指定正整数值设置的时间量（以秒为单位）内，代理继续尝试恢复停止的传输。 例如，`transferRecoveryTimeout=21600` 表示代理会在进入恢复后六小时内一直尝试恢复传输。 此参数的最大值为 `999999999`.	-1

表 91: 高级代理程序属性: 用户出口例程

属性名	描述	缺省值
agentForceConsistentPathDelimiters	将提供给传输出口的源文件和目标文件信息中的路径定界符强制为 UNIX 样式: 正斜杠 (/)。 有效选项为 `true` 和 `false`。	false
destinationTransferEndExitClasses	指定一个实现目标传输用户出口例程的以逗号分隔的类列表。	没有缺省值
destinationTransferStartExitClasses	指定一个实现目标传输开始用户出口例程的以逗号分隔的类列表。	没有缺省值
exitClassPath	指定一个特定于平台、字符分隔的目录列表，充当用户出口例程的类路径。 将在此类路径中的任何条目之前搜索代理的出口目录。	代理的 `exits` 目录
exitNativeLibraryPath	指定一个特定于平台、字符分隔的目录列表，充当用户出口例程的本机库路径。	代理的 `exits` 目录
ioMaxRecordLength	面向记录的文件支持的最大记录长度（字节）。 Managed File Transfer 支持写入具有任意记录长度的面向记录文件。 但是，较大的记录长度可能会导致内存不足错误，因此为了避免这些错误，缺省情况下最大记录长度限制为 64 K。 从面向记录的文件中读取时，整个记录必须适合单个传输块，因此记录长度还受到传输块大小的限制。 该属性仅用于 I/O 用户出口的面向记录文件。	64KB
monitorExitClasses	指定实现了监视器出口例程的类的逗号分隔列表。 有关更多信息，请参阅 MFT 资源监视器用户出口。	没有缺省值
protocolBridgeCredentialExitClasses	指定实现了协议网桥凭证用户出口例程的类的逗号分隔列表。 有关更多信息，请参阅使用出口类映射文件服务器的凭证。	没有缺省值。
sourceTransferEndExitClasses	指定一个实现源传输结束出口例程的以逗号分隔的类列表。	没有缺省值
sourceTransferStartExitClasses	指定一个实现源传输开始出口例程的以逗号分隔的类列表。	没有缺省值
IOExitClasses	指定实现了 I/O 用户出口例程的类的逗号分隔列表。 仅列出实现了 IOExit 接口的类，即不列出实现了其他 I/O 用户出口接口的类，例如，IOExitResourcePath 和 IOExitChannel。 有关更多信息，请参阅使用 MFT 传输 I/O 用户出口。	没有缺省值。

表 92: 高级代理程序属性: *IBM MQ* 客户机压缩

属性名	描述	缺省值
agentDataCompression	仅针对客户机连接支持此属性。 以逗号分隔的与远程 IBM MQ 服务器协商的文件数据传输的压缩类型列表。 您可以在以下主题中查找有关这些压缩类型的信息：消息数据压缩列表。 将检查这些值的有效性，然后按外观顺序将其作为属性传递到代理程序客户机通道。 然后，IBM MQ 客户机处理该客户机通道和远程服务器通道之间的协商，以寻找两个通道上压缩属性之间匹配的最低共同特性。 如果未发现匹配项，那么始终选择 MQCOMPRESS_NONE。	MQCOMPRESS_NONE
agentHeaderCompression	仅针对客户机连接支持此属性。 以逗号分隔的与远程 IBM MQ 服务器协商的头数据传输的压缩类型列表。 接受的值为 MQCOMPRESS_NONE 或 MQCOMPRESS_SYSTEM。 有关这些压缩类型的信息，请参阅以下主题：HdrCompList[2] (MQLONG)。 将检查这些值的有效性，然后按外观顺序将其作为属性传递到代理程序客户机通道。 然后，IBM MQ 客户机处理该客户机通道和远程服务器通道之间的协商，以寻找两个通道上压缩属性之间匹配的最低共同特性。 如果未发现匹配项，那么始终选择 MQCOMPRESS_NONE。	MQCOMPRESS_NONE

z/OS

表 93: 高级代理程序属性: 特定于 *z/OS*

属性名	描述	缺省值
adminGroup	安全管理器组。 该组成员可以: • 使用 **fteStartAgent** 命令启动代理。 • 使用 **fteStopAgent** 命令停止代理。 • 使用 **fteSetAgentTraceLevel** 命令打开或关闭代理程序的跟踪。 • 使用 **fteSetAgentLogLevel** 命令打开或关闭代理程序的日志 • 通过运行指定有 **-d** 参数的 **fteShowAgentDetails** 命令来显示本地代理的详细信息。 定义安全管理器组（例如，*MFTADMIN*），然后将已启动任务用户标识和管理员 TSO 标识添加到该组。 编辑代理属性文件并将 **adminGroup** 属性设置为该安全管理器组的名称。 `adminGroup=MFTADMIN`	None

表 93: 高级代理程序属性: 特定于 z/OS (继续)		
属性名	**描述**	**缺省值**
z/OS bpxwdynAllocAdditionalOptions	Managed File Transfer 使用 BPXWDYN 文本接口创建和打开 z/OS 数据集。 当 BPXWDYN 用于数据集分配时，缺省情况下，Managed File Transfer 可能确保安装数据设备（不是基于磁盘的数据集所需，而是磁带数据集所需）。 由于某些环境可能不支持选项，因此请使用该属性更改此行为。 另外，传输到数据集时， 还可以在命令行指定 BPXWDYN 的选项；这些选项是对该属性指定的那些选项的补充。 使用 `agent.properties` 文件中的 **bpxwdynAllocAdditionalOptions** 属性时，不得指定某些 BPXWDYN 选项。 要获取这些属性的列表，请参阅不得与 MFT 配合使用的 BPXWDYN 属性。	缺省值如下: • 对于 z/OS V1R8 和更高版本为 MOUNT
armELEMTYPE	可选属性。 如果 Automatic Restart Manager (ARM) 将代理配置为重新启动，那么请将此属性设置为关联的 ARM 策略中指定的 ARM ELEMTYPE 参数值。 对于代理，请将 ELEMTYPE 设置为 SYSBFGAG。	未设定
armELEMENT	可选属性。 如果 Automatic Restart Manager (ARM) 将代理配置为重新启动，那么请将此属性设置为关联的 ARM 策略中指定的 ARM ELEMENT 参数值。 您可以将 ELEMENT 值设置为与代理名称相对应。	未设定

表 94: 高级代理程序属性: 其他属性		
属性名	**描述**	**缺省值**
z/OS z/OS legacyXMLMessageMQMDFormat	通过代理生成的 Managed File Transfer XML 消息（例如，日志和传输进度消息）现在将发送至具有空白 MQMD 格式字段的队列。 先前版本的产品将 MQMD 格式字段设置为 MQSTR（文本消息字符串）。 如果将此属性设置为 true，那么就能够将通过代理生成的 Managed File Transfer XML 消息发送至 MQMD 格式字段为 MQSTR 的队列。 **注:** 将使用与对应的命令请求匹配的消息格式发送针对命令的代理应答消息。 如果将 MQMD 格式字段设置为 MQSTR，并且如果在启用了数据转换的 MQ 网络中存在通道，那么 Managed File Transfer 命令 XML 消息可能会损坏。	false
adjustScheduleTimeForDaylightSaving	如果您的企业每天运行调度的传输，因为调度的传输是使用以下命令创建的: • **-oi** 参数设置为 days，并且 • **-tb** 参数已设置为源 例如，在 fteCreateTransfer 命令中，如果将此属性设置为*"true"*，则当时钟前进一小时时，计划的传输时间将前移一小时，而当时钟后退一小时时，计划的传输时间将后移一小时。 例如，如果调度的传输将在凌晨 1:00 运行，那么当时钟前进时，传输将在凌晨 2:00 运行，当时钟后退时，传输将还原为凌晨 1:00。	true

相关概念

多平台上的 MFT 配置选项

恢复中文件传输的超时选项

MFT 沙箱

相关任务

在集群中为多个通道配置 MFT 代理

为多个通道配置 MFT 代理：非集群

相关参考

第 184 页的『MFT 的 Java 系统属性』

必须将一些 Managed File Transfer 命令和代理属性定义为 Java 系统属性，因为它们为无法使用命令或代理属性机制的早期功能定义配置。

MFT 的 SSL/TLS 属性

第 170 页的『MFT command.properties 文件』

`command.properties` 文件指定发出命令时要连接的命令队列管理器以及 Managed File Transfer 联系该队列管理器所需的信息。

第 167 页的『MFT coordination.properties 文件』

`coordination.properties` 文件指定到协调队列管理器的连接详细信息。 由于一些 Managed File Transfer 安装可能共享同一协调队列管理器，因此您可以将符号链接用于共享驱动器上的通用 `coordination.properties` 文件。

第 173 页的『MFT logger.properties 文件』

Managed File Transfer 记录器具有一组配置属性。 在 `logger.properties` 文件中指定这些属性，该文件位于 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name* 目录中。

fteCreateAgent
fteCreateBridgeAgent
fteCreateCDAgent
第 145 页的『在 MFT 属性中使用环境变量』
可以在表示文件或目录位置的 Managed File Transfer 属性中使用环境变量。 这允许在运行产品部分时使用的文件或目录的位置因当前环境而异 (例如，运行命令的用户)。

MFT coordination.properties 文件

`coordination.properties` 文件指定到协调队列管理器的连接详细信息。 由于一些 Managed File Transfer 安装可能共享同一协调队列管理器，因此您可以将符号链接用于共享驱动器上的通用 `coordination.properties` 文件。

`coordination.properties` 文件是由安装程序或 **fteSetupCoordination** 命令创建的。 您可以使用具有 **-f** 标志的 **fteSetupCoordination** 命令更改此文件中的基本协调队列管理器属性。 要更改或添加高级协调队列管理器属性，您必须在文本编辑器中编辑此文件。

`coordination.properties` 文件位于 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* 目录中。

MFT `coordination.properties` 文件包含以下值:

表 *95:* 基本协调队列管理器属性

属性名	描述	缺省值
coordinationCredentialsKeyFile	包含加密凭证时使用的凭证密钥的文件的名称。	没有缺省值的字符串属性。
coordinationQMgr	协调队列管理器的名称。	没有缺省值
coordinationQMgrHost	协调队列管理器的主机名或 IP 地址。	没有缺省值
coordinationQMgrPort	用于以客户机方式连接协调队列管理器的端口号。	1414
coordinationQMgrChannel	用于与协调队列管理器连接的 SVRCONN 通道名称。	SYSTEM.DEF.SVRCONN

如果没有为 coordinationQMgrHost 属性指定值，那么缺省情况下使用绑定方式。

如果您为 coordinationQMgrHost 属性指定值，但未为 coordinationQMgrPort 和 coordinationQMgrChannel 属性指定值，那么缺省情况下使用端口号 1414 和通道 SYSTEM.DEF.SVRCONN。

以下是 `coordination.properties` 文件内容的示例:

```
coordinationQMgr=ERIS
coordinationQMgrHost=kuiper.example.com
coordinationQMgrPort=2005
coordinationQMgrChannel=SYSTEM.DEF.SVRCONN
```

在此示例中， ERIS 是位于系统 `kuiper.example.com` 上的 IBM MQ 队列管理器的名称。 队列管理器 ERIS 是 Managed File Transfer 向其发送日志信息的队列管理器。

高级协调属性

Managed File Transfer 还提供了更高级的协调属性。 如果要使用以下任何属性，请手动编辑 `coordination.properties` 文件以添加所需的高级属性。 括号、逗号 (,) 和反斜线 (\) 是 MFT 命令中的特殊字符，必须用反斜线 (\) 字符转义。 Windows Windows 上的文件路径可以使用双反斜线 (\\) 作为分隔符，也可以使用单个正斜线 (/) 指定。 有关 Java 属性文件中的字符转义的更多信息，请参阅 Oracle 文档属性类的 贾瓦多克。

- 代理属性
- 代码页属性
- 连接属性
- 多实例队列管理器属性

- 队列属性
- 安全属性
- SSL 属性
- 预订属性

表 *96:* 高级协调属性*:* 代理程序

属性名	描述	缺省值
agentStatusJitterTolerance	消息视为过期之前，代理状态消息发布延迟的最大时间量。 该值度量单位为毫秒。 状态消息的期限取决于在协调队列管理器上发布的时间。 但是，代理会在协调队列管理器接收消息之前的一段时间发出消息，以腾出在 IBM MQ 网络中传输所需的时间。 如果该传输所用时间始终相同，那么每隔 60 秒创建的消息，每隔 60 秒就会发布，而与实际传输时间无关。 但是，如果消息之间的传输时间不同，那么可能每隔 60 秒创建消息，但每隔诸如 61、59、58 和 62 秒等时间发布消息。 60 的最大偏差为抖动（在该示例中为 2 秒）。 该属性确定消息视为过期之前，由于抖动造成的最大延迟。	3000

表 *97:* 高级协调属性*:* 代码页

属性名	描述	缺省值
coordinationCcsid	命令连接协作队列管理器所使用的代码页。 代理还使用该代码页执行至协调队列管理器的任何发布。 如果为 coordinationCcsid 指定值，那么还必须为 coordinationCcsidName 指定值。	1208
coordinationCcsidName	coordinationCcsid 的 Java 表示。 如果为 coordinationCcsidName 指定值，那么还必须为 coordinationCcsid 指定值。	UTF8

表 *98:* 高级协调属性*:* 连接

属性名	描述	缺省值
javaLibraryPath	当以绑定方式连接到队列管理器时， Managed File Transfer 必须具 有对 IBM MQ Java 绑定库 的访问权。 缺省情况下， Managed File Transfer 在 IBM MQ 定义的缺省位置中查找绑定库。 如果绑定库位于不同位置，那么请使用该属性以指定绑定库的位置。	*MQ_INSTALLATION_PATH*/java/lib

表 *99:* 高级协调属性*:* 多实例队列管理器

属性名	描述	缺省值
coordinationQMgrStandby	针对通过 coordinationQMgr 属性定义的多实例协调队列管理器的备用实例，用于客户机连接的主机名和端口号，采用 IBM MQ CONNAME 格式。 例如，*host_name*(*port_number*)	没有缺省值

表 *100:* 高级协调属性*:* 队列

属性名	描述	缺省值
dynamicQueuePrefix	该属性定义 IBM MQ 前缀，以用于生成临时队列名称。 dynamicQueuePrefix 属性的格式与 "IBM MQMQOD 结构的 "**DynamicQName** 字段格式相同。 有关更多信息，请参阅创建动态队列。 如果要将特定 IBM MQ 前缀用于需要代理程序响应的命令生成的临时应答队列，那么还可以在 command.properties 文件中定义此属性。	WMQFTE.*
modelQueueName	该属性定义 IBM MQ 模型队列，以用于生成临时队列。 如果您要将特定的 IBM MQ 模型队列用于由需要代理响应的命令生成的临时应答队列，那么还可以在 command.properties 文件中定义此属性。 有关更多信息，请参阅 第 170 页的『MFT command.properties 文件』。	SYSTEM.DEFAULT.MODEL.QUEUE

表 *101:* 高级协调属性*:* 安全性

属性名	描述	缺省值
userIdForClientConnect	经客户机连接流至 IBM MQ 的用户标识。 如果指定了 *java*，那么 JVM 报告的用户名会作为 IBM MQ 连接请求的一部分流动。 该属性的值可以为 None 或 java。	None

表 101: 高级协调属性: 安全性 (继续)		
属性名	**描述**	**缺省值**
coordinationQMgrAuthenticationCredentialsFile	包含用于连接到协调队列管理器的 MQ 连接凭证的文件的路径。	z/OS 有关创建认证凭证文件的详细信息，请参阅 在 z/OS 上配置 MQMFTCredentials.xml。 ALW 请参阅 配置 MQMFTCredentials.xml，以获取有关此文件的位置和许可权的信息。 ALW 有关创建认证凭证文件的更多详细信息，请参阅 MFT 和 IBM MQ 连接认证

表 102: 高级协调属性: *SSL/TLS*		
属性名	**描述**	**缺省值**
coordinationSslCipherSpec	指定在命令和协调队列管理器之间交换数据时，使用的协议、散列算法和加密算法，以及加密密钥中使用的位数。 coordinationSslCipherSpec 的值为 CipherSpec 名称。 该 CipherSpec 名称与协调队列管理器通道中使用的 CipherSpec 名称相同。 Java 的 "IBM MQ 类中的 SSL/TLS "CipherSpecs 和 "CipherSuites"和 "JMS 的 "IBM MQ 类中的 SSL/TLS "CipherSpecs 和 "CipherSuites"中包含有效的 "CipherSpec 名称列表。 coordinationSslCipherSpec 类似于 coordinationSslCipherSuite。 如果指定了 coordinationSslCipherSuite 和 coordinationSslCipherSpec，那么使用 coordinationSslCipherSpec 的值。	None
coordinationSslCipherSuite	指定命令和协调队列管理器交换数据方式的 SSL 方面。 coordinationSslCipherSuite 的值为 CipherSuite 名称。 CipherSuite 名称将映射到代理队列管理器通道中使用的 CipherSpec 名称。 有关更多信息，请参阅 CipherSuite 和 CipherSpec 名称映射。 coordinationSslCipherSuite 类似于 coordinationSslCipherSpec。 如果指定了 coordinationSslCipherSuite 和 coordinationSslCipherSpec，那么使用 coordinationSslCipherSpec 的值。	None
coordinationSslPeerName	指定必须与协调队列管理器提供的名称相匹配的专有名称框架。 专有名称用于检查连接时协调队列管理器提供的标识证书。	None
coordinationSslTrustStore	指定命令信任的证书的位置。 coordinationSslTrustStore 的值为文件路径。括号、逗号 (,) 和反斜线 (\) 是 MFT 命令中的特殊字符，必须用反斜线 (\) 字符转义。 Windows Windows 上的文件路径可以使用双反斜线 (\\) 作为分隔符，也可以使用单个正斜线 (/) 指定。 从 IBM WebSphere MQ 7.5 或更高版本开始，此属性的值可以包含环境变量。	None
coordinationSslTrustStoreType	您要使用的 SSL 密钥库的类型。 支持 JKS 和 PKCS#12 密钥库。 该属性的值可以为 `jks` 或 `pkcs12`。	jks
coordinationSslTrustStoreCredentialsFile	包含 coordinationSslTrustStore 凭证的文件的路径。 此属性的值可以包含环境变量。	此属性的缺省值为 `%USERPROFILE%/MQMFTCredentials.xml` on Windows 和 `$HOME/MQMFTCredentials.xml` on other 平台。
coordinationSslKeyStore	指定命令专用密钥的位置。 coordinationSslKeyStore 的值为文件路径。括号、逗号 (,) 和反斜线 (\) 是 MFT 命令中的特殊字符，必须用反斜线 (\) 字符转义。 Windows Windows 上的文件路径可以使用双反斜线 (\\) 作为分隔符，也可以使用单个正斜线 (/) 指定。 仅当协调队列管理器需要客户机认证时才需要该属性。 此属性的值可以包含环境变量。	None
coordinationSslKeyStoreType	您要使用的 SSL 密钥库的类型。 支持 JKS 和 PKCS#12 密钥库。 该属性的值可以为 `jks` 或 `pkcs12`。	jks
coordinationSslKeyStoreCredentialsFile	包含 coordinationSslKeyStore 凭证的文件的路径。 此属性的值可以包含环境变量。	此属性的缺省值为 `%USERPROFILE%/MQMFTCredentials.xml` on Windows 和 `$HOME/MQMFTCredentials.xml` on other 平台。
coordinationSslFipsRequired	指定要在协调队列管理器级别启用 FIPS 支持。 该属性的值可以为 `true` 或 `false`。 更多信息，请参阅 FIPS 支持 MFT。	false

表 103: 高级协调属性: 预订		
属性名	**描述**	**缺省值**
coordinationSubscriptionTopic	使用此属性指定 SYSTEM.FTE 以外要预订的主题，以便获取有关 IBM MQ 网络状态的出版物。 所有工具仍会发布至 SYSTEM.FTE 主题，但您可以更改自己的 IBM MQ 拓扑，以根据内容将这些出版物发布至不同的主题。 然后，您可以使用此功能强制工具预订其他主题中的一个主题。	SYSTEM.FTE

相关概念

多平台上的 MFT 配置选项

相关参考

fteSetupCoordination

MFT 的 SSL/TLS 属性

第 149 页的『MFT agent.properties 文件』

每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。 `agent.properties` 文件还可以包含更改代理行为的属性。

第 170 页的『MFT command.properties 文件』

`command.properties` 文件指定发出命令时要连接的命令队列管理器以及 Managed File Transfer 联系该队列管理器所需的信息。

第 173 页的『MFT logger.properties 文件』

Managed File Transfer 记录器具有一组配置属性。 在 `logger.properties` 文件中指定这些属性，该文件位于 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name* 目录中。

MFT command.properties 文件

`command.properties` 文件指定发出命令时要连接的命令队列管理器以及 Managed File Transfer 联系该队列管理器所需的信息。

`command.properties` 文件是由安装程序或 **fteSetupCommands** 命令创建的。 您可以使用具有 **-f** 标志的 **fteSetupCommands** 命令更改此文件中的基本命令队列管理器属性。 要更改或添加高级命令队列管理器属性，您必须在文本编辑器中编辑此文件。

某些 Managed File Transfer 命令连接到代理队列管理器或协调队列管理器，而不是命令队列管理器。 有关哪些命令连接到哪个队列管理器的信息，请参阅 哪个 MFT 命令连接到哪个队列管理器。

`command.properties` 文件位于 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name* 目录中。

MFT `command.properties` 文件包含以下值:

表 104: 基本命令队列管理器属性		
属性名	**描述**	**缺省值**
connectionCredentialsKeyFile	包含加密凭证时使用的凭证密钥的文件的名称。	没有缺省值的字符串属性。
connectionQMgr	用于连接到 IBM MQ 网络的队列管理器的名称。	没有缺省值
connectionQMgrHost	连接队列管理器的主机名或 IP 地址。	没有缺省值
connectionQMgrPort	用于以客户机方式连接到连接队列管理器的端口号。	1414
connectionQMgrChannel	用于与连接队列管理器连接的 SVRCONN 通道名称。	SYSTEM.DEF.SVRCONN

如果没有为 connectionQMgrHost 属性指定值，那么缺省情况下使用绑定方式。

如果您为 connectionQMgrHost 属性指定值，但没有为 connectionQMgrPort 和 connectionQMgrChannel 属性指定值，那么缺省情况下使用端口号 1414 和通道 SYSTEM.DEF.SVRCONN。

以下是 `command.properties` 文件内容的示例:

```
connectionQMgr=PLUTO
connectionQMgrHost=kuiper.example.com
connectionQMgrPort=1930
connectionQMgrChannel=SYSTEM.DEF.SVRCONN
```

在此示例中， PLUTO 是位于系统 kuiper.example.com 上的 IBM MQ 队列管理器的名称。 队列管理器 PLUTO 是 Managed File Transfer 命令连接的队列管理器。

高级命令属性

Managed File Transfer 还提供了更高级的命令属性。 如果要使用以下任何属性，请手动编辑 command.properties 文件以添加所需的高级属性。 括号、逗号 (,) 和反斜线 (\) 是 MFT 命令中的特殊字符，必须用反斜线 (\) 字符转义。 Windows Windows 上的文件路径可以使用双反斜线 (\\) 作为分隔符，也可以使用单个正斜线 (/) 指定。 有关 Java 属性文件中的字符转义的更多信息，请参阅 Oracle 文档 属性类的 贾瓦多克。

- 代理属性
- 代码页属性
- 多实例队列管理器属性
- 队列属性
- 安全属性
- SSL 属性

表 *105:* 高级命令属性: 代理程序

属性名	描述	缺省值
failCleanAgentWithNoArguments	缺省情况下，此属性的值为 true，这表示如果仅指定代理名称参数，那么 **fteCleanAgent** 命令将运行失败。 将此属性设置为 false 表示，如果仅设置代理名称参数，那么 **fteCleanAgent** 命令的行为与指定 **-all** 参数的情况下的行为相同。	true

表 *106:* 高级命令属性: 代码页

属性名	描述	缺省值
connectionCcsid	命令连接命令队列管理器所使用的代码页。 如果为 connectionCcsid 指定值，那么还必须为 connectionCcsidName 指定值。	1208
connectionCcsidName	connectionCcsid 的 Java 表示。 如果为 connectionCcsidName 指定值，那么还必须为 connectionCcsid 指定值。	UTF8

表 *107:* 高级连接属性: 多实例队列管理器

属性名	描述	缺省值
connectionQMgrStandby	由 connectionQMgr 属性定义的多实例命令队列管理器的备用实例在建立客户机连接时所使用的主机名和端口号，采用 IBM MQ CONNAME 格式。 例如，*host_name*(*port_number*)	没有缺省值

表 *108:* 高级命令属性: 队列

属性名	描述	缺省值
dynamicQueuePrefix	对于需要代理响应的命令，该属性定义 IBM MQ 前缀以用于生成临时应答队列名称。 dynamicQueuePrefix 属性的格式与 "IBM MQMQOD 结构的 "**DynamicQName** 字段格式相同。 有关更多信息，请参阅创建动态队列。 如果要将特定 IBM MQ 前缀用于 WMQFTE 生成的临时队列，那么还可以在 coordination.properties 文件中定义此属性。	WMQFTE.*
modelQueueName	对于需要代理响应的命令，该属性定义 IBM MQ 模型队列以用于生成临时应答队列。 如果您要将特定 IBM MQ 模型队列用于由 WMQFTE 生成的临时队列，那么您还可以在 coordination.properties 文件中定义该属性。 有关更多信息，请参阅 第 167 页的『MFT coordination.properties 文件』。	SYSTEM.DEFAULT.MODEL.QUEUE
连接属性:		
javaLibraryPath	当以绑定方式连接到队列管理器时， Managed File Transfer 必须具 有对 IBM MQ Java 绑定库 的访问权。 缺省情况下， Managed File Transfer 在 IBM MQ 定义的缺省位置中查找绑定库。 如果绑定库位于不同位置，那么请使用该属性以指定绑定库的位置。	/opt/mqm/java/lib

表 108: 高级命令属性: 队列 (继续)

属性名	描述	缺省值
z/OS z/OS legacyXMLMessageMQMDFormat	Managed File Transfer 命令 XML 消息现在将发送至具有空白 MQMD 格式字段的队列。 先前版本的产品将 MQMD 格式字段设置为 MQSTR（文本消息字符串）。 如果将此属性设置为 true，那么就能够将 Managed File Transfer 命令 XML 消息发送至 MQMD 格式字段为 MQSTR 的队列。 如果将 MQMD 格式字段设置为 MQSTR，并且如果在启用了数据转换的 MQ 网络中存在通道，那么 Managed File Transfer 命令 XML 消息可能会损坏。	false

表 109: 高级命令属性: 安全性

属性名	描述	缺省值
userIdForClientConnect	经客户机连接流至 IBM MQ 的用户标识。 如果指定了 *java*，那么 JVM 报告的用户名会作为 IBM MQ 连接请求的一部分流动。 该属性的值可以为 None 或 java。	None
connectionQMgrAuthenticationCredentialsFile	包含用于连接到命令队列管理器的 MQ 连接凭证的文件的路径。	请参阅 MFT 和 IBM MQ 连接认证 及其子主题。

表 110: 高级命令属性: *SSL/TLS*

属性名	描述	缺省值
connectionSslCipherSpec	指定在命令和命令队列管理器之间交换数据时，使用的协议、散列算法和加密算法，以及加密密钥中使用的位数。 connectionSslCipherSpec 的值为 CipherSpec 名称。 该 CipherSpec 名称与命令队列管理器通道中使用的 CipherSpec 名称相同。 "、"Java 的 "IBM MQ 类中的 SSL/TLS "CipherSpecs 和 "CipherSuites"和 "JMS 的 "IBM MQ 类中的 SSL/TLS "CipherSpecs 和 "CipherSuites 中包含有效的 "CipherSpec 名称列表。 connectionSslCipherSpec 类似于 connectionSslCipherSuite。 如果指定了 connectionSslCipherSuite 和 connectionSslCipherSpec，那么使用 connectionSslCipherSpec 的值。	None
connectionSslCipherSuite	指定命令和命令队列管理器交换数据方式的 SSL 方面。 connectionSslCipherSuite 的值为 CipherSuite 名称。 CipherSuite 名称将映射到代理队列管理器通道中使用的 CipherSpec 名称。 有关更多信息，请参阅 CipherSuite 和 CipherSpec 名称映射。 connectionSslCipherSuite 类似于 connectionSslCipherSpec。 如果指定了 connectionSslCipherSuite 和 connectionSslCipherSpec，那么使用 connectionSslCipherSpec 的值。	None
connectionSslPeerName	指定必须与命令队列管理器提供的名称相匹配的专有名称框架。 专有名称用于检查连接时命令队列管理器提供的标识证书。	None
connectionSslTrustStore	指定命令信任的证书的位置。 connectionSslTrustStore 的值为文件路径。括号、逗号 (,) 和反斜线 (\) 是 MFT 命令中的特殊字符，必须用反斜线 (\) 字符转义。 Windows Windows 上的文件路径可以使用双反斜线 (\\) 作为分隔符，也可以使用单个正斜线 (/) 指定。 此属性的值可以包含环境变量。	None
connectionSslTrustStoreType	您要使用的 SSL 信任库的类型。 支持 JKS 和 PKCS#12 密钥库。 该属性的值可以为 jks 或 pkcs12。	jks
connectionSslTrustStoreCredentialsFile	包含 connectionSslTrustStore 凭证的文件的路径。 此属性的值可以包含环境变量。	此属性的缺省值为 %USERPROFILE%/MQMFTCredentials.xml on Windows 和 $HOME/MQMFTCredentials.xml on other 平台。
connectionSslKeyStore	指定命令专用密钥的位置。 connectionSslKeyStore 的值为文件路径。括号、逗号 (,) 和反斜线 (\) 是 MFT 命令中的特殊字符，必须用反斜线 (\) 字符转义。 Windows Windows 上的文件路径可以使用双反斜线 (\\) 作为分隔符，也可以使用单个正斜线 (/) 指定。 仅当命令代理队列管理器需要客户机认证时才需要该属性。 此属性的值可以包含环境变量。	None
connectionSslKeyStoreType	您要使用的 SSL 密钥库的类型。 支持 JKS 和 PKCS#12 密钥库。 该属性的值可以为 jks 或 pkcs12。 此属性的值可以包含环境变量。	jks

表 110: 高级命令属性: SSL/TLS (继续)		
属性名	**描述**	**缺省值**
connectionSslKeyStoreCredentialsFile	包含 connectionSslKeyStore 凭证的文件的路径。 此属性的值可以包含环境变量。	此属性的缺省值为 %USERPROFILE%/MQMFTCredentials.xml on Windows 和 $HOME/MQMFTCredentials.xml on other 平台。
connectionSslFipsRequired	指定要在命令队列管理器级别启用 FIPS 支持。 该属性的值可以为 true 或 false。 更多信息，请参阅 FIPS 支持 MFT。	false

相关概念

多平台上的 MFT 配置选项

相关参考

第 184 页的『MFT 的 Java 系统属性』
必须将一些 Managed File Transfer 命令和代理属性定义为 Java 系统属性，因为它们为无法使用命令或代理属性机制的早期功能定义配置。

MFT 的 SSL/TLS 属性

第 149 页的『MFT agent.properties 文件』
每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。 `agent.properties` 文件还可以包含更改代理行为的属性。

第 167 页的『MFT coordination.properties 文件』
`coordination.properties` 文件指定到协调队列管理器的连接详细信息。 由于一些 Managed File Transfer 安装可能共享同一协调队列管理器，因此您可以将符号链接用于共享驱动器上的通用 `coordination.properties` 文件。

第 173 页的『MFT logger.properties 文件』
Managed File Transfer 记录器具有一组配置属性。 在 `logger.properties` 文件中指定这些属性，该文件位于 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name* 目录中。

fteSetupCommands：创建 MFT command.properties 文件

fteCleanAgent：清除 MFT 代理

MFT logger.properties 文件

Managed File Transfer 记录器具有一组配置属性。 在 `logger.properties` 文件中指定这些属性，该文件位于 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name* 目录中。

可以在表示文件或目录位置的某些 Managed File Transfer 属性中使用环境变量。 这允许在运行产品的各个部分时使用的文件或目录位置因环境更改而变化，例如哪个用户在运行进程。 有关更多信息，请参阅 第 145 页的『在 MFT 属性中使用环境变量』。

注: 括号、逗号 (,) 和反斜线 (\) 是 MFT 命令中的特殊字符，必须用反斜线 (\) 字符转义。 Windows Windows 上的文件路径可以使用双反斜线 (\\) 作为分隔符，也可以使用单个正斜线 (/) 指定。 有关 Oracle 中 Java 属性文件中字符转义的更多信息，请参阅 Javadoc 以了解 Properties 类。

MFT `logger.properties` 文件包含以下值:

- 第 173 页的『绑定方式连接属性』
- 第 179 页的『客户机方式 SSL/TLS 连接属性』

绑定方式连接属性

表 111: logger.properties 文件的绑定方式连接属性		
属性名	**描述**	**缺省值**
wmqfte.logger.type	正在使用的记录器类型：文件或数据库。 将该值设置为 FILE 或 DATABASE。	无缺省值

表 111: *logger.properties* 文件的绑定方式连接属性 (继续)

属性名	描述	缺省值
wmqfte.max.transaction.messages	落实事务之前在该事务中处理的消息的最大数目。 在循环日志记录方式下，队列管理器可用于实时数据的空间容量是固定的。 确保您将该属性设置为足够低的值，以便可用空间不会耗尽。	50
wmqfte.max.transaction.time	前后两次事务落实之间所经过的最大时间长度（毫秒）。	5000
wmqfte.max.consecutive.reject	可连续拒绝的最大消息数（即未遇到有效消息）。 如果超过该数量，那么记录器将认定问题不在于消息自身，而在于配置。 例如，如果您使数据库中某个代理名称列的宽度小于所有代理名称，那么引用代理的所有消息都将被拒绝。	50
wmqfte.reject.queue.name	记录器将自己无法处理的消息放置到的队列的名称。 如果您有数据库记录器，请参阅 MFT 记录器错误处理和拒绝，以获取可能将哪些消息放入此队列的详细信息。	SYSTEM.FTE.LOG.RJCT.*logger_name*
wmqfte.command.queue.name	记录器从中读取控制其行为的命令消息的队列的名称。	SYSTEM.FTE.LOG.CMD.*logger_name*
wmqfte.queue.manager	记录器连接到的队列管理器。 此参数是必需的，并且是与队列管理器的绑定方式连接所需的所有参数。 (有关用于连接到远程队列管理器的属性，请参阅 第 179 页的表 112。)	无缺省值
wmqfte.message.source.type	以下某个值: **automatic subscription** 缺省值。 记录器在 SYSTEM.FTE/Log/# 中定义的队列管理器上创建并使用其自己的持久的受管预订。 这是适合大多数场景的值。 **administrative subscription** 如果自动预订不合适，那么您可以定义其他预订（例如，通过使用 IBM MQ Explorer、MQSC 或 PCF），并指示记录器使用该预订。 例如，使用该值将日志空间分区，以便一个记录器处理代理 A 到 H，另一个记录器处理 I 到 P，第三个记录器处理 Q 到 Z。 **queue** 如果 IBM MQ 拓扑表示不方便创建记录器的预订，那么您可以改为使用队列。 配置 IBM MQ，使该队列接收通常由协调队列管理器上的 SYSTEM.FTE/Log/# 预订接收的消息。	自动预订
wmqfte.message.source.name	如果消息源类型是**管理预订**或**队列**，那么使用预订或队列的名称。 如果源类型是**自动预订**，那么忽略该属性。	无缺省值
wmqfte.database.credentials.file	包含用于连接至数据库的用户名和密码的文件。 此属性的值可以包含环境变量。 有关更多信息，请参阅 MFT 凭证文件格式。	z/OS 有关创建认证凭证文件的信息，请参阅 在 z/OS 上配置 MQMFTCredentials.xml。 ALW 有关此文件的位置和许可权的信息，请参阅 配置 MQMFTCredentials.xml。 ALW 另请参阅 MFT 和 IBM MQ 连接认证 (connection authentication)。

表 111: *logger.properties* 文件的绑定方式连接属性 (继续)

属性名	描述	缺省值
wmqfte.database.driver	数据库的 JDBC 驱动程序类的位置。 这通常是 JAR 文件的路径和文件名。 例如，Db2 on AIX 系统的 2 类驱动程序需要 `/opt/IBM/db2/V9.5/java/db2jcc.jar` 文件。 在 Windows 系统上，将路径分隔符指定为正斜杠字符 (/)，例如 `C:/Program Files/IBM/SQLLIB/java/db2jcc.jar`。 在 z/OS 上，指定 `db2jcc.jar` 文件的完整路径。 例如，`wmqfte.database.driver=/db2/db2v10/jdbc/classes/db2jcc.jar`。 在 z/OS 系统上，必须引用以下所有 JAR 文件： • `db2jcc.jar` • `db2jcc_license_cisuz.jar` • `db2jcc_javax.jar` 如果数据库驱动程序包含多个 JAR 文件（例如，Db2 V9.1 需要一个驱动程序 JAR 文件和一个许可证 JAR 文件），请将所有这些 JAR 文件包含在此属性中。 通过针对平台使用类路径分隔符（即，在 Windows 系统上为分号字符 (;)，在其他平台上为冒号字符 (:)），分隔多个文件名。	无缺省值
wmqfte.database.exclude .duplicate. metadata	控制是否在元数据表中存储条目来包含可在数据库记录器模式内的其他表中找到的信息。 将此值设置为 `true` 或 `false`。 缺省情况下，不再存储这些元数据条目，因为这与现有数据重复，因此会浪费数据库存储容量。 属性条目以及表（其中显示相同数据）如下所示： • com.ibm.wmqfte.SourceAgent TRANSFER_EVENT 或 CALL_REQUEST • com.ibm.wmqfte.DestinationAgent TRANSFER_EVENT • com.ibm.wmqfte.MqmdUser TRANSFER_EVENT 或 CALL_REQUEST • com.ibm.wmqfte.OriginatingUser TRANSFER_EVENT 或 CALL_REQUEST • com.ibm.wmqfte.OriginatingHost TRANSFER_EVENT 或 CALL_REQUEST • com.ibm.wmqfte.TransferId TRANSFER 或 CALL_REQUEST • com.ibm.wmqfte.JobName TRANSFER 或 CALL_REQUEST 将此属性的值设置为 `false` 会使这些元数据条目存储在元数据表中。	true

表 111: *logger.properties* 文件的绑定方式连接属性 (继续)

属性名	描述	缺省值
wmqfte.database.host	仅限 Db2： 要使用类型 4 JDBC 驱动程序连接到的数据库服务器的主机名。 如果指定了此属性的值，那么也必须指定 `wmqfte.database.port` 的值。 如果这两个属性都没有定义，那么数据库记录器将使用缺省类型 2 JDBC 驱动程序进行连接。 如果指定了此属性的值，那么必须具有此记录器的凭证文件（文件路径由 wmqfte.database.credentials.file 属性定义），并且必须能够访问该文件以定义用于连接至数据库的用户名和密码，即使数据库位于本地系统上也是如此。	无缺省值
wmqfte.database.name	包含 Managed File Transfer 日志表的数据库实例 (或使用 Db2 for z/OS 时的子系统) 的名称。	无缺省值
wmqfte.database.type	正在使用的数据库管理系统：Db2 或 Oracle。 将该值设置为 db2 或 `oracle`。	db2
wmqfte.database.port	仅限 Db2： 要使用类型 4 JDBC 驱动程序连接到的数据库服务器的端口号。 如果指定了此属性的值，那么也必须指定 `wmqfte.database.host` 的值。 如果这两个属性都没有定义，那么数据库记录器将使用缺省类型 2 JDBC 驱动程序进行连接。 如果指定了此属性的值，那么必须具有此记录器的凭证文件（文件路径由 wmqfte.database.credentials.file 属性定义），并且必须能够访问该文件以定义用于连接至数据库的用户名和密码，即使数据库位于本地系统上也是如此。	无缺省值
wmqfte.database.schema	仅限 Db2： 包含 Managed File Transfer 日志记录表的数据库模式。 在大多数情况下缺省值都适用，但您可能需要出于自身特定于站点的数据库考虑，指定其他值。	FTELOG
wmqfte.database.native.library.path	包含所选数据库驱动程序（如果有）所需的本机库的路径。 AIX 例如，Db2 on AIX 系统的 2 类驱动程序需要 `/opt/IBM/db2/V9.5/lib32/` 中的库。 作为该属性的替代方案，您可以使用其他方法设置 java.library.path 系统属性。	无缺省值
wmqfte.file.logger.fileDirectory	文件记录器日志文件所在的目录。	`mqft/logs/`*`coordination_dir`*`/loggers/`*`logger_name`*`/logs`
wmqfte.file.logger.fileSize	允许一个日志文件增大到的最大大小。 该大小值是大于零的正整数，后跟以下某个单位：KB、MB、GB、m（分钟）、h（小时）、d（天）、w（周）。 例如， `wmqfte.file.logger.fileSize=5MB` 指定最大文件大小 5MB，`wmqfte.file.logger.fileSize=2d` 指定最大文件大小 2 天的数据。	10MB
wmqfte.file.logger.fileCount	要创建的日志文件的最大数目。 数据量超过此数目的文件中可以存储的最大量时，将删除最旧的文件，以便文件数从不超过指定的值。	3

表 *111: logger.properties* 文件的绑定方式连接属性 *(*继续*)*

属性名	描述	缺省值
wmqfte.file.logger.mode	正在使用的记录器方式：circular 或 linear。 将该值设置为 CIRCULAR 或 LINEAR。 CIRCULAR - 文件记录器将信息写入文件，直至该文件达到其最大大小（使用 **wmqfte.file.logger.fileSize** 属性定义）。 达到最大大小时，文件记录器将启动新文件。 在该方式下写入了信息的文件的最大数量由使用 **wmqfte.file.logger.fileCount** 属性定义的值来控制。 达到该最大文件数时，文件记录器将删除首个文件并重新创建该文件以用作当前活动文件。 如果 **wmqfte.file.logger.fileSize** 属性中定义的值为固定大小的字节单位（例如，KB、MB 或 GB），那么该方式下使用的磁盘空间的上限等于 fileSize 乘以 fileCount。 如果 **wmqfte.file.logger.fileSize** 属性中定义的值为时间单位（例如，m、h、d 或 w），那么最大大小将取决于这些时间段内系统中的日志消息吞吐量。 在该方式下运行时使用的日志文件命名约定为：*logger_namenumber-timestamp*.log，其中： • *logger_name* 是在 **fteCreateLogger** 命令中为记录器指定的名称。 • *number* 是文件在集合内的编号。 • *timestamp* 是创建文件时的时间戳记。 例如，LOGGER1-20111216123430147.log LINEAR - 文件记录器将信息写入文件，直至该文件达到其最大大小（使用 **wmqfte.file.logger.fileSize** 属性定义）。 达到最大大小时，文件记录器将启动新文件。 先前已写入信息的文件将不会删除，从而允许将其保留为日志消息的历史记录。 在 linear 方式下运行时不会删除文件，从而将忽略 **wmqfte.file.logger.fileCount** 属性，这是因为可创建的文件数量没有上限。 因为在该方式下运行时没有上限，所以有必要跟踪日志文件所用的磁盘空间量以避免磁盘空间不足。 在该方式下运行时使用的日志文件命名约定为：*logger_name-timestamp*.log，其中： • *logger_name* 是在 **fteCreateLogger** 命令中为记录器指定的名称。 • *timestamp* 是创建文件时的时间戳记。 例如，LOGGER-20111216123430147.log	无缺省值
wmqfte.max.retry.interval	记录器遇到持续性错误时，两次重试之间的最长时间（秒）。 某些错误情况（例如，丢失数据库连接）会使记录器无法继续运行。 当出现这种类型的情况时，记录器将回滚当前事务，等待一段时间，然后重试。 记录器等待的时间最初很短，这样短暂错误可以快速得以解决。 但是，记录器每次重试时，其等待的时间都会延长。 这可防止在错误情况持续较长时间时（例如，数据库关闭进行维护时）执行过多的不必要工作。 使用此属性可设置等待时间长度的限制，以便在解决错误情况的合理时间内进行重试。	600

表 111: *logger.properties* 文件的绑定方式连接属性 (继续)

属性名	描述	缺省值
immediateShutdownTimeout	记录器等待任何未完成的操作正常完成和关闭的时间(以秒计)。 缺省情况下，记录器等待 10 秒以完成操作。 如果在超时之前未完成操作，那么记录器会将以下事件消息写入 `output0.log` 并结束。 `BFGDB0082I: The logger is ending immediately.` 如果指定值为零，那么记录器将等待任何时间来完成当前操作。 如果 **immediateShutdownTimeout** 的值设置为小于零，那么将使用缺省值。 此属性同时适用于独立数据库记录器和文件类型记录器。	10
loggerCredentialsKeyFile	包含加密凭证时使用的凭证密钥的文件的名称。	没有缺省值的字符串属性。
loggerQMgrRetryInterval	记录器的进程控制器对队列管理器的可用性进行检查的时间间隔（秒）。	30
maxRestartCount	在 maxRestartInterval 属性值所指定的时间间隔内可发生的最大重新启动次数。 当超出该值时，记录器的进程控制器会停止重新启动记录器，而改为执行基于 maxRestartDelay 属性值的操作。	4
maxRestartInterval	一段时间间隔（秒），记录器的进程控制器度量此时间间隔内记录器的重新启动次数。 如果该时间间隔内的重新启动次数超过 maxRestartCount 属性的值，那么记录器的进程控制器会停止重新启动记录器。 记录器的进程控制器会改为执行基于 maxRestartDelay 属性值的操作。	120
maxRestartDelay	确定当记录器重新启动的频率超过 maxRestartCount 和 maxRestartInterval 属性值时，记录器的进程控制器的行为。 如果指定小于或等于零的值，那么记录器的进程控制器会停止。 如果指定大于零的值，那么这是重置记录器的进程控制器保存的重新启动历史记录信息并重新启动记录器之前等待的秒数。	-1
wmqfte.oracle.port	记录器用于连接到 Oracle 实例的端口。 此端口也称为 TNS 侦听器。	1521
wmqfte.oracle.host	记录器用于连接到 Oracle 实例的主机。	localhost
armELEMTYPE	可选属性。 如果 Automatic Restart Manager (ARM) 将记录器配置为重新启动，那么请将此属性设置为关联的 ARM 策略中指定的 ARM ELEMTYPE 参数值。 对于记录器，请将 ELEMTYPE 设置为 SYSBFGLG。	未设定
armELEMENT	可选属性。 如果 Automatic Restart Manager (ARM) 将记录器配置为重新启动，那么请将此属性设置为关联的 ARM 策略中指定的 ARM ELEMENT 参数值。 您可以将 ELEMENT 值设置为与记录器名称相对应。	未设定
loggerQMgrAuthenticationCredentialsFile	包含用于连接到记录器的协调队列管理器的 MQ 连接凭证的文件的路径。	z/OS 有关创建认证凭证文件的信息，请参阅 在 z/OS 上配置 MQMFTCredentials.xml。 ALW 有关此文件的位置和许可权的信息，请参阅 配置 MQMFTCredentials.xml。 ALW 另请参阅 MFT 和 IBM MQ 连接认证 (connection authentication)。

表 111: logger.properties 文件的绑定方式连接属性 (继续)		
属性名	**描述**	**缺省值**
跟踪	可选属性。 要在记录器启动时启用跟踪的情况下运行记录器时的跟踪规范。 跟踪规范是以逗号分隔的类、等于字符以及跟踪级别列表。 例如， `com.ibm.wmqfte.databaselogger` 和 `com.ibm.wmqfte.databaselogger.operation=all` 您可以在冒号分隔的列表中指定多个跟踪规范。 例如， `com.ibm.wmqfte.databaselogger=moderate:com.ibm.wmqfte.databaselogger.operation=all`	None
traceFiles	可选属性。 保存的跟踪文件的总数。 该值适用于记录器的进程控制器以及记录器自身。	5
traceSize	可选属性。 将跟踪合并到下一个文件之前，每个跟踪文件的最大大小（MB）。 该值适用于记录器的进程控制器以及记录器自身。	20
wmqfte.file.logger.filePermissions	可选属性。 用于指定记录器的日志文件所需的许可权类型。 该属性既适用于线性日志，也适用于圆形日志，其值可以是 *UserReadWriteOnly* 或 *UserReadWriteAllRead*. 该 *UserReadWriteOnly* 值的现有等效权限为 600，而 *UserReadWriteAllRead* 值的等效权限为 644。 许可权中的任何更改都适用于新创建的记录器文件。 如果为属性输入值无效，那么记录器将采用缺省值并向输出日志发出消息 BFGDB0083W 。	UserReadWriteOnly

客户机方式 SSL/TLS 连接属性

支持客户机方式连接到使用 SSL/TLS 的记录器队列管理器所需的属性。

表 112: logger.properties 文件的客户机方式 SSL/TLS 连接属性		
属性名	**描述**	**缺省值**
wmqfte.queue.manager.host	记录器队列管理器的主机名或 IP 地址。	无缺省值
wmqfte.queue.manager.port	记录器队列管理器正在侦听的端口。	1414
wmqfte.queue.manager.channel	记录器队列管理器上服务器连接通道的名称。	SYSTEM.DEF.SVRCONN
wmqfte.Ssl.CipherSuite	指定记录器和记录器队列管理器交换数据方式的 TLS 方面。 **wmqfte.Ssl.CipherSuite** 的值为 CipherSuite 名称。 CipherSuite 名称将映射到记录器队列管理器通道中使用的 CipherSpec 名称。 有关更多信息，请参阅 CipherSuite 和 CipherSpec 名称映射。	无缺省值
wmqfte.Ssl.PeerName	指定必须与记录器队列管理器提供的名称相匹配的专有名称框架。 专有名称用于检查连接时队列管理器提供的标识证书。	无缺省值

表 112: logger.properties 文件的客户机方式 SSL/TLS 连接属性 (继续)		
属性名	**描述**	**缺省值**
wmqfte.Ssl.TrustStore	指定记录器信任的证书的位置。 **wmqfte.Ssl.TrustStore** 的值为文件路径。 括号、逗号 (,) 和反斜线 (\) 是 MFT 命令中的特殊字符，必须用反斜线 (\) 字符转义。 Windows Windows 上的文件路径可以使用双反斜线 (\\) 作为分隔符，也可以使用单个正斜线 (/) 指定。 请注意，此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.TrustStoreCredentialsFile	包含 **wmqfte.Ssl.TrustStore** 凭证的文件的路径。 请注意，此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.TrustStoreType	您要使用的 SSL 密钥库的类型。 支持 JKS 和 PKCS#12 密钥库。 该属性的值可以为 jks 或 pkcs12。	jks
wmqfte.Ssl.KeyStore	指定记录器的专用密钥的位置。 **wmqfte.Ssl.KeyStore** 的值是文件路径。 括号、逗号 (,) 和反斜线 (\) 是 MFT 命令中的特殊字符，必须用反斜线 (\) 字符转义。 Windows Windows 上的文件路径可以使用双反斜线 (\\) 作为分隔符，也可以使用单个正斜线 (/) 指定。 请注意，此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.KeyStore.CredentialsFile	包含 **wmqfte.Ssl.KeyStore** 凭证的文件的路径。 请注意，此属性的值可以包含环境变量。	无缺省值
wmqfte.Ssl.KeyStoreType	您要使用的 SSL 密钥库的类型。 支持 JKS 和 PKCS#12 密钥库。 该属性的值可以为 jks 或 pkcs12。	jks
wmqfte.Ssl.FipsRequired	指定要在记录仪级别启用 FIPS 支持。 该属性的值可以为 true 或 false。 有关详细信息，请参阅 FIPS 支持 MFT。	false

相关概念

MFT 的 SSL/TLS 属性

相关参考

第 145 页的『在 MFT 属性中使用环境变量』
可以在表示文件或目录位置的 Managed File Transfer 属性中使用环境变量。 这允许在运行产品部分时使用的文件或目录的位置因当前环境而异 (例如，运行命令的用户)。

第 149 页的『MFT agent.properties 文件』
每个 Managed File Transfer Agent 都有自己的属性文件 `agent.properties`，其中必须包含代理用于连接到其队列管理器的信息。 `agent.properties` 文件还可以包含更改代理行为的属性。

第 170 页的『MFT command.properties 文件』
`command.properties` 文件指定发出命令时要连接的命令队列管理器以及 Managed File Transfer 联系该队列管理器所需的信息。

第 167 页的『MFT coordination.properties 文件』
`coordination.properties` 文件指定到协调队列管理器的连接详细信息。 由于一些 Managed File Transfer 安装可能共享同一协调队列管理器，因此您可以将符号链接用于共享驱动器上的通用 `coordination.properties` 文件。

由 LogTransfer 函数生成的输出

传输日志事件捕获从提交传输到完成之前的传输进度的详细信息。 还会捕获有关传输进入再同步的信息，以帮助您了解传输的进度。

传输事件格式

传输事件采用 JSON 格式并写入 `transferlogN.json` 文件，该文件是在代理程序的日志目录中创建的，其中 `N` 是一个以 `0` 为缺省值的数字。 每个事件都包含以下公共属性:

- 日期和时间 (UTC)
- 唯一标识

根据事件类型和传输日志级别，写入的事件信息中包含其他属性。 当传输日志级别 *info* 写入最少信息时，*verbose* 级别包含更详细的信息。 以下 第 181 页的『样本事件』 部分描述了代理程序记录的传输事件的一些示例。

唯一标识

包含唯一标识是为了帮助您在传输过程中轻松识别不同阶段，例如 `BFGTL0001`。 唯一标识是 **eventDescription** 属性的一部分，由两个部分组成:

BFGTL
用于所有标识的前缀，其中 BFG 是 Managed File Transfer 中使用的标准后缀， TL 指示这是传输日志。

号码
从 1 开始的唯一编号。 例如:

```
{
  "eventDescription": "BFGTL0001: New transfer request submitted"
}
```

样本事件

下表将某些事件描述为附加函数记录的信息的示例。 表 日志级别 的第二列指示记录事件的级别。

要点: 如果 **logTransfer** 级别设置为 详细 或 中等，那么以下属性将包含在事件信息中:

- **sourceAgent**
- **destinationAgent**
- **threadId**

事件	日志级别	描述

要转移的项目列表	verbose	{
		"dateTime": "<Data time in UTC>",
		"eventDescription": "BFGTL0002I: Generated detailed transfer item list.",
		"destinationAgent": "<Name of destination agent>",
		"sourceAgent": "<Name of source agent>",
		"threadId": "0000001d",
		"totalItemsInTransfer": <Number of items in the transfer>,
		"transferId": "<Transfer Identifier>",
		"transferItemsList": [{"source":"source item name",
		"destination": "destination item name"}]
		}
		
		Example:
		{
		"dateTime": "2022-01-14T12:56:54.219Z UTC",
		"eventDescription": "BFGTL0002I: Generated detailed transfer item list.",
		"destinationAgent": "QMBAGQ",
		"sourceAgent": "QMBAG1",
		"threadId": "0000001d",
		"totalItems": 1,
		"transferId": "414d5120514d412020202020202020202063bd17610a390040",
		"transferItems": [{
		"destination": "/results/rts/target/destFile.txt",
		"source": "DESTINATIONQ@QMB"
		}]
		}

事件	日志级别	描述

源代理记录的传输的审计信息	参考，中等，详细	<pre>{
		  "dateTime": "<Date and time in UTC>",
		  "transferId": "<Transfer Id>",
		  "eventDescription": "BFGTL0008: Transfer progress information ",
		  "progressInformation": [{
		  "destination": "<Destination item name>",
		  "resultCode": "<Result of transfer>",
		  "supplement": "<Any supplemental information.>"
		  "source": "<Source item name>"
		  }],
		  "sourceAgent": "<Source agent name>",
		  "destinationAgent": "<Destination agent name>",
		  "threadId": "Thread Id",
		}
		
		Example:
		{
		  "dateTime": "2022-01-14T14:17:20.075Z UTC",
		  "transferId": "414D5120514D412020202020202020202FD01761013B0040",
		  "eventDescription": "BFGTL0008: Transfer progress information",
		  "progressInformation": [{
		  "source": "MONITORQ@QMA"
		  "destination": "/results/rts/target/destinationFile.txt",
		  "resultCode": "Successful",
		  "supplement": ""
		  }],
		  "sourceAgent": "QMAAG1",
		  "destinationAgent": "QMBAG1",
		  "threadId": "00000023",
		}</pre>
		以下是记录的信息示例 (如果级别为 `info`):
		<pre>{
		  "dateTime":"2022-01-16T03:28:26.234Z UTC",
		"transferId":"414D51204D4654514D20202020202020EB89416101760040",
		"eventDescription":"BFGTL0008I: Transfer progress information",
		  "progressInformation": {
		  "warnings":0,
		  "failed":1,
		  "successful":0,
		  "unknown":0}
		}</pre>
		以下是记录的信息示例 (如果级别为 `moderate`):
		<pre>{
		"dateTime":"2022-01-16T03:30:46.766Z UTC",
		"transferId":"414D51204D4654514D20202020202020EB89416101850040",
		"eventDescription":"BFGTL0008I: Transfer progress information",
		"progressInformation": [{
		"source":"FileZilla:/home/mft/*",
		"destination":"C:/MFT/FileZilla:/home/mft/*",
		"itemSize":0,
		"resultCode":"Failed",
		"supplement":"BFGIO0110E: File \"/home/mft/*\" does not exist."}],
		"sourceAgent":"IMQFT01"
		"destinationAgent":"SRC",
		}</pre>

打开项时发生错误	info	<pre>{
		  "dateTime": "<Date and time in UTC>",
		  "transferId": "<Transfer Id>",
		  "eventDescription": "BFGTL0054E: Failed to open an item to transfer
		data.",
		  "itemName": "Name of file or queue",
		  "error":"<Error description>"
		  "sourceAgent": "<Source agent name>",
		  "destinationAgent": "<Destination agent name>",
		  "threadId": "<Thread Id>",
		}
		
		Example:
		{
		  "dateTime": "2022-01-14T14:42:24.902Z UTC",
		  "transferId": "414D512043514D485830312020202020B0D4176101370040",
		  "eventDescription": " BFGTL0054E: Failed to open an item to transfer
		data.",
		  "itemName":"/inbound/resource.xml",
		  "error": "File not found"
		  "sourceAgent": "CQMHX01AG1",
		  "destinationAgent": "HX01AG1",
		  "threadId": "0000001c"</pre>

事件	日志级别	描述

要在代理启动时恢复的传输的列表	verbose	<pre>{
		  "dateTime": "<Date and time in UTC>",
		  "eventDescription": "The list of transfers being recovered as part
		of agent recovery process.",
		  "agentName": "<Agent name>",
		  "transfers": [{"transferId":"<transfer state>"}]
		  "threadId": "<Thread Id>",
		}
		
		Example:
		{
		  "dateTime": "2022-01-14T14:42:24.902Z UTC",
		  "eventDescription": "The list of transfers being recovered as part
		of agent recovery process.",
		  "agentName": "CQMHX01AG1",
		  "transfers":
		[{"414D512043514D485830312020202020B0D4176101370040":"completeReceived
		"},
		{"414D512043514D485830312020202020B0D4176101370050":"resynchronizing"}]
		  "threadId": "0000001c",
		}</pre>

相关参考

第 184 页的『MFT 的 Java 系统属性』
必须将一些 Managed File Transfer 命令和代理属性定义为 Java 系统属性，因为它们为无法使用命令或代理属性机制的早期功能定义配置。

fteCreateAgent

第 145 页的『在 MFT 属性中使用环境变量』
可以在表示文件或目录位置的 Managed File Transfer 属性中使用环境变量。 这允许在运行产品部分时使用的文件或目录的位置因当前环境而异 (例如，运行命令的用户)。

MFT 的 Java 系统属性

必须将一些 Managed File Transfer 命令和代理属性定义为 Java 系统属性，因为它们为无法使用命令或代理属性机制的早期功能定义配置。

通过定义环境变量 BFG_JVM_PROPERTIES，为要运行 Managed File Transfer 命令的 JVM 定义系统属性和其他 JVM 选项。 例如，要在 UNIX 类平台上设置 com.ibm.wmqfte.maxConsoleLineLength 属性，请将该变量定义如下：

```
export BFG_JVM_PROPERTIES="-Dcom.ibm.wmqfte.maxConsoleLineLength=132"
```

如果要将代理程序作为 Windows 服务运行，那么可以通过在 **fteModifyAgent** 命令上指定 -sj 参数来修改代理程序的 Java 系统属性。

表 *113: Java* 系统属性

属性名	描述	值
com.ibm.wmqfte.maxConsoleLineLength	可以写入控制台的最大行长度。 超过该长度的行会自动换行。 该值用字节表示（不是字符）。	IBM i 对于 IBM i 的缺省长度为 132 字节。 ALW 对于 AIX, Linux, and Windows，长度不受限制。 z/OS 对于 z/OS，长度不受限制。

表 113: Java 系统属性 (继续)		
属性名	**描述**	**值**
com.ibm.wmqfte.daemon.windows.windowsServiceLogFilesm	（仅限 Windows。） 指定要保存的 Windows 服务日志文件的最大数量。 如果这些应用程序作为 Windows 服务运行，会在代理和数据库记录器日志目录中创建 Windows 服务日志文件。 Windows 服务日志文件以 *service* 为前缀命名，并包含关于服务启动和停止的消息。	5

相关概念
多平台上的 MFT 配置选项
使用 MFT 的提示与技巧

MFT 的 SHA-2 CipherSpec 和 CipherSuite

Managed File Transfer 支持 SHA-2 CipherSpec 和 CipherSuite。

有关可用于代理程序与 IBM MQ 队列管理器之间的连接的 CipherSpecs 和 CipherSuites 的更多信息，请参阅 IBM MQ classes for Java 中的 TLS CipherSpecs 和 CipherSuites 和 针对 JMS 的 IBM MQ 类中的 SSL/TLS CipherSpecs 和 CipherSuites。

有关配置 CipherSpecs 和 CipherSuites 以用于协议网桥代理 (PBA) 和 FTPS 服务器的更多信息，请参阅 FTPS 服务器支持协议网桥 和 协议网桥属性文件格式。

如果要符合 SP 800-131A 标准，必须满足以下要求：

- 必须使用正确配置的 FTPS；不支持 SFTP。
- 远程服务器必须仅发送符合 SP 800-131A 标准的密码套件。

相关概念
MFT 的 SSL/TLS 属性

MFT 文件记录器配置文件

除了 `logger.properties` 文件，Managed File Transfer 独立文件记录器的配置目录中还有 XML 配置文件。 此配置文件称为 `FileLoggerFormat.xml`，它定义由该文件记录器用于将消息写入日志文件的格式。 该文件的内容必须符合 `FileLoggerFormat.xsd` 文件中定义的 XML 模式。

相关概念
MFT 独立文件记录器格式

相关参考
第 173 页的『MFT logger.properties 文件』
Managed File Transfer 记录器具有一组配置属性。 在 `logger.properties` 文件中指定这些属性，该文件位于 *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name* 目录中。

第 185 页的『MFT 独立文件记录器缺省日志格式』
Managed File Transfer 独立文件记录器的缺省日志文件格式定义

第 190 页的『独立文件记录器格式 XSD』
独立文件格式的模式。

MFT 独立文件记录器缺省日志格式

Managed File Transfer 独立文件记录器的缺省日志文件格式定义

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="1.00" xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <callCompleted>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/transaction/action/@time</insert>
```

```
          <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="3"  ignoreNull="false">/transaction/status/@resultCode</insert>
          <insert type="user"   width="0"  ignoreNull="false">/transaction/agent/@agent</insert>
          <insert type="user"   width="0"  ignoreNull="false">/transaction/agent/@QMgr</insert>
          <insert type="user"   width="0"  ignoreNull="false">/transaction/job/name</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/transferSet/call/command/
@type</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/transferSet/call/command/
@name</insert>
          <insert type="system" width="0"  ignoreNull="true">callArguments</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/transferSet/call/callResult/
@outcome</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/transferSet/call/callResult/
result/error</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </callCompleted>
    <callStarted>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/transaction/action/@time</insert>
          <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="0"  ignoreNull="false">/transaction/agent/@agent</insert>
          <insert type="user"   width="0"  ignoreNull="false">/transaction/agent/@QMgr</insert>
          <insert type="user"   width="0"  ignoreNull="false">/transaction/job/name</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/transferSet/call/command/
@type</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/transferSet/call/command/
@name</insert>
          <insert type="system" width="0"  ignoreNull="true">callArguments</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </callStarted>
    <monitorAction>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/monitorLog/action/@time</insert>
          <insert type="user"   width="48" ignoreNull="false">/monitorLog/@referenceId</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="3"  ignoreNull="false">/monitorLog/status/@resultCode</insert>
          <insert type="user"   width="0"  ignoreNull="false">/monitorLog/@monitorName</insert>
          <insert type="user"   width="0"  ignoreNull="false">/monitorLog/monitorAgent/@agent</insert>
          <insert type="user"   width="0"  ignoreNull="false">/monitorLog/monitorAgent/@QMgr</insert>
          <insert type="user"   width="0"  ignoreNull="false">/monitorLog/action</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </monitorAction>
    <monitorCreate>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/monitorLog/action/@time</insert>
          <insert type="user"   width="48" ignoreNull="false">/monitorLog/@referenceId</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="0"  ignoreNull="false">/monitorLog/@monitorName</insert>
          <insert type="user"   width="0"  ignoreNull="false">/monitorLog/monitorAgent/@agent</insert>
          <insert type="user"   width="0"  ignoreNull="false">/monitorLog/monitorAgent/@QMgr</insert>
          <insert type="user"   width="0"  ignoreNull="false">/monitorLog/action</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </monitorCreate>
    <monitorFired>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/monitorLog/action/@time</insert>
          <insert type="user"   width="48" ignoreNull="false">/monitorLog/@referenceId</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="3"  ignoreNull="false">/monitorLog/status/@resultCode</insert>
          <insert type="user"   width="0"  ignoreNull="false">/monitorLog/@monitorName</insert>
          <insert type="user"   width="0"  ignoreNull="false">/monitorLog/monitorAgent/@agent</insert>
          <insert type="user"   width="0"  ignoreNull="false">/monitorLog/monitorAgent/@QMgr</insert>
          <insert type="user"   width="0"  ignoreNull="false">/monitorLog/action</insert>
          <insert type="user"   width="48" ignoreNull="false">/monitorLog/references/taskRequest</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </monitorFired>
```

```
    <notAuthorized>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/notAuthorized/action/@time</insert>
          <insert type="user"   width="48" ignoreNull="false">/notAuthorized/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="3"  ignoreNull="false">/notAuthorized/status/@resultCode</insert>
          <insert type="user"   width="12" ignoreNull="false">/notAuthorized/action</insert>
          <insert type="user"   width="12" ignoreNull="false">/notAuthorized/authority</insert>
          <insert type="user"   width="0"  ignoreNull="false">/notAuthorized/originator/userID</insert>
          <insert type="user"   width="0"  ignoreNull="false">/notAuthorized/status/supplement</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </notAuthorized>
    <scheduleDelete>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/schedulelog/action/@time</insert>
          <insert type="user"   width="48" ignoreNull="false">/schedulelog/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="3"  ignoreNull="false">/schedulelog/status/@resultCode</insert>
          <insert type="user"   width="0"  ignoreNull="false">/schedulelog/sourceAgent/@agent</insert>
          <insert type="user"   width="12" ignoreNull="false">/schedulelog/action</insert>
          <insert type="user"   width="0"  ignoreNull="false">/schedulelog/originator/userID</insert>
          <insert type="user"   width="0"  ignoreNull="true">/schedulelog/status/supplement</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </scheduleDelete>
    <scheduleExpire>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/schedulelog/action/@time</insert>
          <insert type="user"   width="48" ignoreNull="false">/schedulelog/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="3"  ignoreNull="false">/schedulelog/status/@resultCode</insert>
          <insert type="user"   width="0"  ignoreNull="false">/schedulelog/sourceAgent/@agent</insert>
          <insert type="user"   width="12" ignoreNull="false">/schedulelog/action</insert>
          <insert type="user"   width="0"  ignoreNull="false">/schedulelog/originator/userID</insert>
          <insert type="user"   width="0"  ignoreNull="true">/schedulelog/status/supplement</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </scheduleExpire>
    <scheduleSkipped>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/schedulelog/action/@time</insert>
          <insert type="user"   width="48" ignoreNull="false">/schedulelog/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="3"  ignoreNull="false">/schedulelog/status/@resultCode</insert>
          <insert type="user"   width="0"  ignoreNull="false">/schedulelog/sourceAgent/@agent</insert>
          <insert type="user"   width="12" ignoreNull="false">/schedulelog/action</insert>
          <insert type="user"   width="0"  ignoreNull="false">/schedulelog/originator/userID</insert>
          <insert type="user"   width="0"  ignoreNull="true">/schedulelog/status/supplement</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </scheduleSkipped>
    <scheduleSubmitInfo>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/schedulelog/action/@time</insert>
          <insert type="user"   width="48" ignoreNull="false">/schedulelog/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="3"  ignoreNull="false">/schedulelog/status/@resultCode</insert>
          <insert type="user"   width="0"  ignoreNull="false">/schedulelog/sourceAgent/@agent</insert>
          <insert type="user"   width="12" ignoreNull="false">/schedulelog/action</insert>
          <insert type="user"   width="0"  ignoreNull="false">/schedulelog/originator/userID</insert>
          <insert type="user"   width="0"  ignoreNull="true">/schedulelog/schedule/submit/</insert>
          <insert type="user"   width="0"  ignoreNull="true">/schedulelog/schedule/submit/@timezone</
insert>
          <insert type="user"   width="3"  ignoreNull="true">/schedulelog/schedule/repeat/frequency</
insert>
          <insert type="user"   width="12" ignoreNull="true">/schedulelog/schedule/repeat/frequency/
@interval</insert>
          <insert type="user"   width="3"  ignoreNull="true">/schedulelog/schedule/repeat/expireCount</
insert>
          <insert type="user"   width="0"  ignoreNull="true">/schedulelog/status/supplement</insert>
        </inserts>
        <separator>;</separator>
```

```
      </format>
    </scheduleSubmitInfo>
    <scheduleSubmitTransfer>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/schedulelog/action/@time</insert>
          <insert type="user"   width="48" ignoreNull="false">/schedulelog/@ID</insert>
          <insert type="system" width="10" ignoreNull="false">type</insert>
          <insert type="user"   width="0"  ignoreNull="false">/transaction/sourceAgent/@agent |
          /transaction/sourceWebUser/@webGatewayAgentName |
          /transaction/sourceWebGateway/@webGatewayAgentName</insert>
          <insert type="user"   width="0"  ignoreNull="false">/transaction/sourceAgent/@QMgr |
         /transaction/sourceWebUser/@webGatewayAgentQMgr |
        /transaction/sourceWebGateway/@webGatewayAgentQMgr</insert>
          <insert type="user"   width="0"  ignoreNull="false">/transaction/destinationAgent/@agent |
        /transaction/destinationWebUser/@webGatewayAgentName |
        /transaction/destinationWebGateway/@webGatewayAgentName</insert>
          <insert type="user"   width="0"  ignoreNull="false">/transaction/destinationAgent/@QMgr |
       /transaction/destinationWebUser/@webGatewayAgentQMgr |
       /transaction/destinationWebGateway/@webGatewayAgentQMgr</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </scheduleSubmitTransfer>
    <scheduleSubmitTransferSet>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/schedulelog/action/@time</insert>
          <insert type="user"   width="48" ignoreNull="false">/schedulelog/@ID</insert>
          <insert type="system" width="10" ignoreNull="false">type</insert>
          <insert type="user"   width="0"  ignoreNull="false">source/file | source/queue</insert>
          <insert type="user"   width="5"  ignoreNull="true">source/@type</insert>
          <insert type="user"   width="6"  ignoreNull="true">source/@disposition</insert>
          <insert type="user"   width="0"  ignoreNull="false">destination/file | destination/queue</
insert>
          <insert type="user"   width="5"  ignoreNull="true">destination/@type</insert>
          <insert type="user"   width="9"  ignoreNull="true">destination/@exist</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </scheduleSubmitTransferSet>
    <transferStarted>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/transaction/action/@time</insert>
          <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="3"  ignoreNull="true">/transaction/status/@resultCode</insert>
          <insert type="user"   width="0"  ignoreNull="false">/transaction/sourceAgent/@agent |
        /transaction/sourceWebUser/@webGatewayAgentName |
        /transaction/sourceWebGateway/@webGatewayAgentName</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/sourceAgent/@QMgr |
       /transaction/sourceWebUser/@webGatewayAgentQMgr |
       /transaction/sourceWebGateway/@webGatewayAgentQMgr</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/sourceAgent/@agentType |
       /transaction/sourceWebUser/@webGatewayAgentType |
       /transaction/sourceWebGateway/@webGatewayAgentType</insert>
          <insert type="user"   width="0"  ignoreNull="false">/transaction/destinationAgent/@agent |
       /transaction/destinationWebUser/@webGatewayAgentName |
       /transaction/destinationWebGateway/@webGatewayAgentName</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/destinationAgent/@QMgr |
       /transaction/destinationWebUser/@webGatewayAgentQMgr |
       /transaction/destinationWebGateway/@webGatewayAgentQMgr</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/originator/userID</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/job/name</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/scheduleLog/@ID</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </transferStarted>
    <transferCancelled>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/transaction/action/@time</insert>
          <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="3"  ignoreNull="true">/transaction/status/@resultCode</insert>
          <insert type="user"   width="0"  ignoreNull="false">/transaction/sourceAgent/@agent |
      /transaction/sourceWebUser/@webGatewayAgentName |
      /transaction/sourceWebGateway/@webGatewayAgentName</insert>
          <insert type="user"   width="0"  ignoreNull="true">/transaction/sourceAgent/@QMgr |
      /transaction/sourceWebUser/@webGatewayAgentQMgr |
```

```
  /transaction/sourceWebGateway/@webGatewayAgentQMgr</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/sourceAgent/@agentType |
  /transaction/sourceWebUser/@webGatewayAgentType |
  /transaction/sourceWebGateway/@webGatewayAgentType</insert>
      <insert type="user"   width="0"  ignoreNull="false">/transaction/destinationAgent/@agent |
 /transaction/destinationWebUser/@webGatewayAgentName |
 /transaction/destinationWebGateway/@webGatewayAgentName</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/destinationAgent/@QMgr |
 /transaction/destinationWebUser/@webGatewayAgentQMgr |
 /transaction/destinationWebGateway/@webGatewayAgentQMgr</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/destinationAgent/@agentType |
/transaction/destinationWebUser/@webGatewayAgentType |
/transaction/destinationWebGateway/@webGatewayAgentType</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/originator/userID</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/job/name</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/status/supplement</insert>
    </inserts>
    <separator>;</separator>
  </format>
</transferCancelled>
<transferComplete>
  <format>
    <inserts>
      <insert type="user"   width="19" ignoreNull="false">/transaction/action/@time</insert>
      <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6"  ignoreNull="false">type</insert>
      <insert type="user"   width="3"  ignoreNull="true">/transaction/status/@resultCode</insert>
      <insert type="user"   width="0"  ignoreNull="false">/transaction/sourceAgent/@agent |
       /transaction/sourceWebUser/@webGatewayAgentName |
      /transaction/sourceWebGateway/@webGatewayAgentName</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/sourceAgent/@QMgr |
      /transaction/sourceWebUser/@webGatewayAgentQMgr |
      /transaction/sourceWebGateway/@webGatewayAgentQMgr</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/sourceAgent/@agentType |
      /transaction/sourceWebUser/@webGatewayAgentType |
      /transaction/sourceWebGateway/@webGatewayAgentType</insert>
      <insert type="user"   width="0"  ignoreNull="false">/transaction/destinationAgent/@agent |
      /transaction/destinationWebUser/@webGatewayAgentName |
      /transaction/destinationWebGateway/@webGatewayAgentName</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/destinationAgent/@QMgr |
      /transaction/destinationWebUser/@webGatewayAgentQMgr |
      /transaction/destinationWebGateway/@webGatewayAgentQMgr</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/destinationAgent/@agentType |
      /transaction/destinationWebUser/@webGatewayAgentType |
      /transaction/destinationWebGateway/@webGatewayAgentType</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/originator/userID</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/job/name</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/status/supplement</insert>
    </inserts>
    <separator>;</separator>
  </format>
</transferComplete>
<transferDelete>
  <format>
    <inserts>
      <insert type="user"   width="19" ignoreNull="false">/transaction/action/@time</insert>
      <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6"  ignoreNull="false">type</insert>
      <insert type="user"   width="3"  ignoreNull="true">/transaction/status/@resultCode</insert>
      <insert type="user"   width="0"  ignoreNull="false">/transaction/sourceAgent/@agent |
      /transaction/sourceWebUser/@webGatewayAgentName |
      /transaction/sourceWebGateway/@webGatewayAgentName</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/sourceAgent/@QMgr |
      /transaction/sourceWebUser/@webGatewayAgentQMgr |
      /transaction/sourceWebGateway/@webGatewayAgentQMgr</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/sourceAgent/@agentType |
      /transaction/sourceWebUser/@webGatewayAgentType |
      /transaction/sourceWebGateway/@webGatewayAgentType</insert>
      <insert type="user"   width="0"  ignoreNull="false">/transaction/destinationAgent/@agent |
      /transaction/destinationWebUser/@webGatewayAgentName |
      /transaction/destinationWebGateway/@webGatewayAgentName</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/destinationAgent/@QMgr |
      /transaction/destinationWebUser/@webGatewayAgentQMgr |
      /transaction/destinationWebGateway/@webGatewayAgentQMgr</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/destinationAgent/@agentType |
      /transaction/destinationWebUser/@webGatewayAgentType |
      /transaction/destinationWebGateway/@webGatewayAgentType</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/originator/userID</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/job/name</insert>
      <insert type="user"   width="0"  ignoreNull="true">/transaction/status/supplement</insert>
    </inserts>
    <separator>;</separator>
```

```
      </format>
    </transferDelete>
    <transferProgress>
      <format>
        <inserts>
          <insert type="user"   width="19" ignoreNull="false">/transaction/action/@time</insert>
          <insert type="user"   width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6"  ignoreNull="false">type</insert>
          <insert type="user"   width="3"  ignoreNull="true">status/@resultCode</insert>
          <insert type="user"   width="0"  ignoreNull="false">source/file | source/queue</insert>
          <insert type="user"   width="0"  ignoreNull="false">source/file/@size | source/queue/@size</
insert>
          <insert type="user"   width="5"  ignoreNull="true">source/@type</insert>
          <insert type="user"   width="6"  ignoreNull="true">source/@disposition</insert>
          <insert type="user"   width="0"  ignoreNull="true">source/file/@alias | source/queue/@alias</
insert>
          <insert type="user"   width="0"  ignoreNull="true">source/file/@filespace | source/queue/
@filespace</insert>
          <insert type="user"   width="0"  ignoreNull="true">source/@correlationBoolean1</insert>
          <insert type="user"   width="0"  ignoreNull="true">source/@correlationNum1</insert>
          <insert type="user"   width="0"  ignoreNull="true">source/@correlationString1</insert>
          <insert type="user"   width="0"  ignoreNull="false">destination/file | destination/queue</
insert>
          <insert type="user"   width="0"  ignoreNull="false">destination/file/@size | destination/queue/
@size</insert>
          <insert type="user"   width="5"  ignoreNull="true">destination/@type</insert>
          <insert type="user"   width="9"  ignoreNull="true">destination/@exist</insert>
          <insert type="user"   width="0"  ignoreNull="true">destination/file/@alias | destination/queue/
@alias</insert>
          <insert type="user"   width="0"  ignoreNull="true">destination/file/@filespace | destination/
queue/@filespace</insert>
          <insert type="user"   width="0"  ignoreNull="true">destination/file/@truncateRecords</insert>
          <insert type="user"   width="0"  ignoreNull="true">destination/@correlationBoolean1</insert>
          <insert type="user"   width="0"  ignoreNull="true">destination/@correlationNum1</insert>
          <insert type="user"   width="0"  ignoreNull="true">destination/@correlationString1</insert>
          <insert type="user"   width="0"  ignoreNull="true">status/supplement</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </transferProgress>
  </messageTypes>
</logFormatDefinition>
```

相关参考

MFT 独立文件记录器格式

第 190 页的『独立文件记录器格式 XSD』

独立文件格式的模式。

独立文件记录器格式 XSD

独立文件格式的模式。

模式

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
@start_non_restricted_prolog@
Version: %Z% %I% %W% %E% %U% [%H% %T%]

Licensed Materials - Property of IBM

5724-H72

Copyright IBM Corp. 2011, 2025. All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
@end_non_restricted_prolog@
 -->

<!--
     This schema defines the format of the FileLoggerFormat XML file that contains the definition
     of the format to use when logging FTE log messages to a file. When an XML file that conforms
     to this schema is processed by a file logger it can contain definitions for one or more
     message type(s) that define how log messages of those types are output to the file log.
```

```
-->

<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">

<xsd:include schemaLocation="fteutils.xsd"/>

    <!--
        Defines the logFileDefinition and version number
        <logFileDefinition version="1.00" ...
            <messageTypes>
                ...
            </messageTypes>
        </logFileDefinition>
     -->
    <xsd:element name="logFileDefinition">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element   name="messageTypes" type="messageTypesType" maxOccurs="1" minOccurs="1"/>
            </xsd:sequence>
            <xsd:attribute name="version"      type="versionType"      use="required"/>
        </xsd:complexType>
    </xsd:element>

    <!--
        Defines the set of accepted message types. The definition of individual message types
        is optional. If a particular types element is present but empty then no line will be
        output for messages of that type. If a particular types element is not present then
        the default format will be used to format messages of that type.
     -->
    <xsd:complexType name="messageTypesType">
         <xsd:sequence>
            <xsd:element name="callCompleted"             type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="callStarted"               type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="monitorAction"             type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="monitorCreate"             type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="monitorFired"              type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="notAuthorized"             type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="scheduleDelete"            type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="scheduleExpire"            type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="scheduleSkipped"           type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="scheduleSubmitInfo"        type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="scheduleSubmitTransfer"    type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="scheduleSubmitTransferSet" type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="transferStarted"           type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="transferCancelled"         type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="transferComplete"          type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="transferDelete"            type="messageType"  maxOccurs="1"
minOccurs="0"/>
            <xsd:element name="transferProgress"          type="messageType"  maxOccurs="1"
minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>

    <!--
        Defines the content of a message type definition e.g.

        <callStarted>
         <format>
          ...
         </format>
        <callStarted>
     -->
    <xsd:complexType name="messageType">
        <xsd:sequence>
            <xsd:element name="format"   type="messageFormatType"  maxOccurs="1" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
```

```
    <!--
        Defines the content of a message format definition e.g.

         <format>
          <inserts>
           ...
          </inserts
          <separator>;</separator>
         </format>
     -->
    <xsd:complexType name="messageFormatType">
        <xsd:sequence>
            <xsd:element name="inserts"   type="insertsType"  maxOccurs="1" minOccurs="1"/>
            <xsd:element name="separator" type="scheduleType" maxOccurs="1" minOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>

    <!--
        Defines the content of the inserts element e.g.

        <inserts>
         <insert ...>
         <insert ...>
         ...
        </inserts>
     -->
    <xsd:complexType name="insertsType">
        <xsd:sequence>
             <xsd:element name="insert" type="insertType" maxOccurs="unbounded" minOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>

    <!--
        Defines the content of an insert definition e.g.

        <insert type="user" width="0" ignoreNull="true">/transaction/@ID</insert>
     -->
    <xsd:complexType name="insertType">
        <xsd:attribute name="type"       type="insertTypeType"         use="required"/>
        <xsd:attribute name="width"      type="xsd:nonNegativeInteger" use="required"/>
        <xsd:attribute name="ignoreNull" type="xsd:boolean"            use="required"/>
    </xsd:complexType>

    <!--
        Defines the accepted choices for the insert type attribute.
     -->
    <xsd:simpleType name="insertTypeType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="user"/>
            <xsd:enumeration value="system"/>
        </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>
```

相关参考

MFT 独立文件记录器格式

第 185 页的『MFT 独立文件记录器缺省日志格式』

Managed File Transfer 独立文件记录器的缺省日志文件格式定义

z/OS SSQFCMD 库

IBM MQ Managed File Transfer for z/OS 提供的 SCSQFCMD 库包含充当作业模板的成员，这些作业可用于创建 Managed File Transfer 配置以及创建和管理代理程序或记录器。

下表显示了库的内容。

成员	描述
BFGCOPY	用于创建 SCSQFCMD 库副本的作业
BFGCUSTM	用于为代理程序或记录器定制库副本的作业
BFGXCROB	**fteObfuscate** 样本模板。

成员	描述
BFGXLGCR	**fteCreateLogger** 模板。
BFGXMNCR	**fteCreateMonitor** 样本模板。
BFGXMNDE	**fteDeleteMonitor** 样本模板。
BFGXPRAN	**fteAnt** 样本模板
BFGXSTDE	**fteDeleteScheduledTransfer** 样本模板
BFGXTMCR	**fteCreateTemplate** 样本模板
BFGXTMDE	**fteDeleteTemplate** 样本模板
BFGXTRCA	**fteCancelTransfer** 样本模板
BFGXTRCR	**fteCreateTransfer** 样本模板
BFGYAGST	用于启动代理程序的启动式任务过程的模板
BFGYLGST	用于启动记录器的启动式任务过程的模板
BFGZAGCL	**fteCleanAgent** 样本模板
BFGZAGCR	**fteCreateAgent** 样本模板
BFGZAGDE	**fteDeleteAgent** 样本模板
BFGZAGLG	**fteSetAgentLogLevel** 样本模板
BFGZAGLI	**fteListAgents** 样本模板
BFGZAGPI	**ftePingAgent** 样本模板
BFGZAGSH	**fteShowAgentDetails** 样本模板
BFGZAGSP	**fteStopAgent** 样本模板
BFGZAGST	**fteStartAgent** 样本模板
BFGZAGTC	**fteSetAgentTraceLevel** 样本模板
BFGZCFCR	**fteSetupCoordination** 样本模板
BFGZCFDF	**fteChangeDefaultConfigurationOptions** 样本模板
BFGZCMCR	**fteSetupCommands** 样本模板
BFGZCMD	数据集中其他成员使用的 REXX 脚本的模板
BFGZLGDE	**fteDeleteLogger** 样本模板
BFGZLGSH	**fteShowLoggerDetails** 样本模板
BFGZLGSP	**fteStopLogger** 样本模板
BFGZLGST	**fteStartLogger** 样本模板
BFGZLGTC	**fteSetLoggerTraceLevel** 样本模板
BFGZMNLI	**fteListMonitors** 样本模板
BFGZPID	**fteSetProductId** 样本模板
BFGZPROF	数据集中其他成员使用的 shell 脚本模板
BFGZPRSH	**fteDisplayVersion** 样本模板
BFGZRAS	**fteRas** 样本模板

成员	描述
BFGZSTLI	**fteListScheduledTransfers** 样本模板
BFGZTMLI	**fteListTemplates** 样本模板

有关如何使用 SCSQFCMD 库生成新库以创建 Managed File Transfer 配置以及创建和管理代理程序或记录器的详细信息，请参阅 配置 Managed File Transfer for z/OS。

相关参考

第 145 页的『在 MFT 属性中使用环境变量』

可以在表示文件或目录位置的 Managed File Transfer 属性中使用环境变量。 这允许在运行产品部分时使用的文件或目录的位置因当前环境而异 (例如，运行命令的用户)。

SYSTEM.FTE 主题

SYSTEM.FTE 主题是关于 Managed File Transfer 用于记录传输并存储有关代理、监视器、调度和模板的协调队列管理器的主题。

主题结构

```
SYSTEM.FTE
      /Agents
            /agent_name
      /monitors
            /agent_name
      /Scheduler
            /agent_name
      /Templates
            /template_ID
      /Transfers
            /agent_name
                  /transfer_ID
      /Log
            /agent_name
                  /Monitors
                  /schedule_ID
                  /transfer_ID
```

SYSTEM.FTE/Agents/*agent_name*

本主题包含描述您的 Managed File Transfer 网络中的代理及其属性的保留出版物。 将定期更新有关本主题的消息以及代理状态。 有关更多信息，请参阅 MFT 代理状态消息格式。

SYSTEM.FTE/monitors/*agent_name*

本主题包含描述与代理 *agent_name* 相关联的资源监视器的保留出版物。 保留出版物的 XML 符合模式 `MonitorList.xsd`。 有关更多信息，请参阅 MFT 监视器列表消息格式。

SYSTEM.FTE/Scheduler/*agent_name*

本主题包含描述与代理 *agent_name* 相关联的所有活动调度的保留出版物。 保留出版物的 XML 符合模式 `ScheduleList.xsd`。 有关更多信息，请参阅 MFT 调度列表消息格式。

SYSTEM.FTE/Templates

本主题包含描述 Managed File Transfer 拓扑中定义的所有模板的保留出版物。

- 与每个模板相关的出版物将发布到名为 `SYSTEM.FTE/Templates/`*`template_ID`* 的子主题。

有关此保留发布内容的示例，请参阅 MFT 示例模板 XML 消息。

SYSTEM.FTE/Transfers/*agent_name*

本主题包含描述源自代理 *agent_name* 的传输状态的出版物。 与每个传输相关的出版物将发布到名为 `SYSTEM.FTE/Transfers/`*`agent_name`*`/`*`transfer_ID`* 的子主题。 IBM MQ Explorer 插件使用这些出版物来提供有关个别传输的进度信息。 出版物的 XML 符合模式 `TransferStatus.xsd`。 有关更多信息，请参阅 文件传输状态消息格式。

SYSTEM.FTE/Log/*agent_name*

本主题包含记录有关源自代理 *agent_name* 的传输、监视器和调度信息的出版物。 这些出版物可由数据库记录器记录，以提供 Managed File Transfer 网络中所发生事件的审计记录。

- 与每个传输相关的出版物将发布到名为 SYSTEM.FTE/Log/*agent_name*/*transfer_ID* 的子主题，并且出版物的 XML 符合模式 TransferLog.xsd。 有关更多信息，请参阅文件传输日志消息格式。
- 与每个调度的传输相关的出版物将发布到名为 SYSTEM.FTE/Log/*agent_name*/*schedule_ID* 的子主题，并且出版物的 XML 符合模式 ScheduleLog.xsd。 有关更多信息，请参阅 调度文件传输日志消息格式。
- 与每次监视相关的出版物将发布到名为 SYSTEM.FTE/Log/*agent_name*/*Monitors*/*monitor_name*/*monitor_ID* 的子主题，并且出版物的 XML 符合模式 MonitorLog.xsd。 有关更多信息，请参阅 MFT 监视器日志消息格式。

MFT 代理队列设置

由 **fteCreateAgent** 命令生成的 MQSC 命令脚本使用设置为以下值的参数创建代理队列。 如果不使用提供用于创建队列的 MQSC 脚本，而是手动创建队列，那么请确保将以下参数设置为给定的值。

代理操作队列

代理的操作队列具有以下名称：

- SYSTEM.FTE.COMMAND.*agent_name*
- SYSTEM.FTE.DATA.*agent_name*
- SYSTEM.FTE.EVENT.*agent_name*
- SYSTEM.FTE.REPLY.*agent_name*
- SYSTEM.FTE.STATE.*agent_name*

表 *114:* 代理操作队列参数

参数	**值（如果适用）**
DEFPRTY	0
DEFSOPT	SHARED
GET	ENABLED
MAXDEPTH	5000
MAXMSGL	4194304
MSGDLVSQ	PRIORITY
PUT	ENABLED
RETINTVL	999999999
SHARE	
NOTRIGGER	
USAGE	NORMAL
REPLACE	

代理权限队列

代理的权限队列具有以下名称：

- SYSTEM.FTE.AUTHADM1.*agent_name*
- SYSTEM.FTE.AUTHAGT1. 代理名称
- SYSTEM.FTE.AUTHMON1.*agent_name*
- SYSTEM.FTE.AUTHOPS1.*agent_name*
- SYSTEM.FTE.AUTHSCH1.*agent_name*

- SYSTEM.FTE.AUTHTRN1.*agent_name*

表 *115:* 代理权限队列参数

参数	值（如果适用）
DEFPRTY	0
DEFSOPT	SHARED
GET	ENABLED
MAXDEPTH	0
MAXMSGL	0
MSGDLVSQ	PRIORITY
PUT	ENABLED
RETINTVL	999999999
SHARE	
NOTRIGGER	
USAGE	NORMAL
REPLACE	

相关参考

fteCreate 代理 (创建 MFT 代理)

MFT 系统队列和系统主题

Managed File Transfer 包含大量系统队列和一个仅限内部使用的系统主题。

名称以 SYSTEM.FTE 是 Managed File Transfer (MFT) 的内部系统队列。 请勿删除这些队列，因为这样做会阻止 IBM MQ MFT 正常工作。 第 196 页的表 116 显示了每个队列上的消息类型:

表 *116:* 队列名称，类型和用法

队列名称	队列类型	用法
SYSTEM.FTE.AUTHAGT1.agent_name	权限	用于配置发送和接收传输请求的权限的队列。
SYSTEM.FTE.AUTHTRN1.agent_name	权限	用于配置启动和取消受管传输的权限的队列。 还要启动受管调用。
SYSTEM.FTE.AUTHMON1.agent_name	权限	用于配置权限以允许用户创建或删除由同一用户创建的资源监视器的队列。
SYSTEM.FTE.AUTHOPS1.agent_name	权限	用于配置权限以删除由另一用户创建的资源监视器和调度传输的队列。
SYSTEM.FTE.AUTHSCH1.agent_name	权限	用于配置权限以创建或删除由同一用户创建的调度传输的队列。
SYSTEM.FTE.AUTHADM1.agent_name	权限	用于配置关闭代理程序的权限的队列，使用 **fteStopAgent** 命令上的 **-m** 选项。
SYSTEM.FTE.COMMAND.agent_name	操作	用于向代理程序发送命令请求的队列。

表 116: 队列名称，类型和用法 (继续)		
队列名称	**队列类型**	**用法**
SYSTEM.FTE.DATA.agent_name	操作	目标代理用于保存源代理发送的数据的队列。
SYSTEM.FTE.REPLY.agent_name	操作	用于从目标代理接收应答的队列。
SYSTEM.FTE.STATE.agent_name	操作	用于保存传输请求状态的队列。
SYSTEM.FTE.EVENT.agent_name	操作	用于保存资源监视器历史记录的队列。
SYSTEM.FTE.HA.agent_name	操作	队列由高可用性代理程序实例用作锁定。

如果代理正在参与消息到文件或文件到消息传输，那么可能需要修改 SYSTEM.FTE.STATE.agent_name 队列的定义以允许执行这些受管传输。 有关此信息的更多信息，请参阅 有关设置与消息大小相关联的 MQ 属性和 MFT 属性的指南。

注意: 不得更改其他系统队列的定义。

此外，请勿修改或删除 SYSTEM.FTE 主题，因为这也仅供内部使用。

临时队列

Managed File Transfer 针对各种用途创建临时队列。 每个队列的名称以 WMQFTE 开头。 缺省情况下。（句点是缺省前缀的一部分。） 如果想要更改此前缀，您可以在 command.properties 文件和/或 coordination.properties 文件中使用 **dynamicQueuePrefix** 属性。 command.properties 文件中的属性用于设置为响应需要代理响应的命令而创建的临时队列的前缀。 coordination.properties 文件中的属性用于设置为其他目的创建的临时队列的前缀; 例如， *WMQFTE*.FTE.TIMECHCK.QUEUE，其中 *WMQFTE*。 是 **dynamicQueuePrefix** 属性定义的值。

相关参考

限制 MFT 代理操作的用户权限

MFT 对象命名约定

请针对 Managed File Transfer 对象使用以下命名约定：

- 代理程序和记录器名称:
  - 最多可包含 28 个字符，并且不区分大小写。
  - 以小写或混合大小写形式输入将转换为大写
  - 必须符合标准 IBM MQ 对象命名约定。

    下面详细描述了这些约定：IBM MQ 对象的命名规则。
- 除了 IBM MQ 对象命名约定外，还有:
  - 不能在代理程序名称或记录器名称中使用正斜杠 (/) 字符
  - 不能在代理程序名称或记录器名称中使用百分比 (%) 字符。
- 属性文件中的属性名称区分大小写。
- 队列管理器名称区分大小写。
- 对于某些平台，文件名区分大小写。
- 资源监视器和传输模板名称:
  - 不区分大小写
  - 以小写或混合大小写形式输入将转换为大写
  - 不得包含星号 (*) ，百分号 (%) 或问号 (?) 字符
- 协议文件服务器名称必须:

- 最小长度为 2 个字符，但没有最大长度限制
- 不区分大小写
- 必须符合标准 IBM MQ 对象命名约定。

  下面详细描述了这些约定：IBM MQ 对象的命名规则。

IBM i 集成文件系统 (IFS) 中的文件

IBM i IFS 中的文件名不能包含以下任何字符：

- 反斜杠 (\)
- 正斜杠 (/)
- 冒号 (:)
- 星号（*）
- 问号（?）
- 引号 (")
- 小于号 (<)
- 大于号 (>)
- 竖线 (|)

如果尝试将名称中包含这些字符的文件传输到 IBM i IFS，那么传输这些文件将失败。

数据集名称

z/OS 数据集具有命名限制，这影响名称最大长度以及可用于数据集名称的可用字符。 PDS 数据集成员名最多可有 8 个字符且不能包含点 (.) 字符。 当传输到数据集时，必须明确指定名称，这意味着这些命名限制不会产生问题。 但当从文件传输到 PDS 成员时，文件路径可能不会映射到 PDS 成员名。 当传输到 PDS 数据集时，每个源文件都将成为 PDS 成员，并且会通过源名称生成每个成员名。

PDS 成员名是 z/OS 非限定名，由以下正则表达式定义：

```
[a-zA-Z$#@][a-zA-Z0-9$#@]{0-7}
```

以下方案用于将源数据集或源文件名转换为有效的 PDS 成员名。 将以所列顺序应用注意事项：

1. 仅使用名称中最后一个正斜杠 (/)、最后一个反斜杠 (\) 或最后一个冒号 (:) 字符后的字符。 即，仅使用文件路径的名称部分。
2. 对于源文件（非数据集或 PDS 成员），将忽略最后一个点号 (.)（含）后面的字符。
3. 对于任何长度超过八个字符的名称，仅使用最后八个字符。
4. 使用 @ 符号字符替换点号字符。
5. 使用 @ 符号字符替换无效字符。
6. 如果转换未生成任何字符，那么 PDS 成员名为 @。

MFT 代理状态消息

高可用性代理程序以 XML 格式发布状态信息。

显示有关三个备用实例的信息的样本 XML

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentStandbyStatus version="6.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="AgentStandbyStatus.xsd">
    <instance host="9.122.123.124" agentVersion="9.1.4.0" />
    <instance host="agenthost.ibm.com" agentVersion="9.1.4.0" />
    <instance host="10.11.12.14" agentVersion="9.1.4.0" />
</AgentStandby>
```

代理程序状态发布，其中嵌入了备用状态 XML。

备用状态 XML 以粗体显示。

```
<?xml version="1.0" encoding="UTF-8"?>
<properties version="1.0">
  <entry key="SourceTransferStates"/>
  <entry key="queueManagerPort">1414</entry>
  <entry key="agentStandbyInstances">&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;AgentStandbyStatus
version="6.00"
        xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="AgentStandbyStatus.xsd"&gt;&lt;Instances&gt;&lt;instance
host="9.122.123.124"
        agentVersion="9.1.4.0" /&gt;&lt;instance host="agenthost.ibm.com" agentVersion="9.1.4.0" /
&gt;&lt;instance host="10.11.12.14"
        agentVersion="9.1.4.0" /&gt;&lt;/Instances&gt;&lt;/AgentStandbyStatus&gt;</entry>
  <entry key="agentType">STANDARD</entry>
  <entry key="agentDeclaredHostName">MFTHA1</entry>
  <entry key="agentDescription"/>
  <entry key="maxQueuedTransfers">1000</entry>
  <entry key="agentTimeZone">America/Los_Angeles</entry>
  <entry key="agentOsName">Windows Server 2019</entry>
  <entry key="PublishTimeUTC">2019-05-22T06:02:50Z</entry>
  <entry key="queueManagerHost">localhost</entry>
  <entry key="AgentStartTimeUTC">2019-05-22T04:13:02Z</entry>
  <entry key="agentTraceLevel">&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;
        agentTraceStatus version="6.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="AgentTraceStatus.xsd"&gt;&lt;trace
        level="all"&gt;com.ibm.wmqfte&lt;/trace&gt;&lt;/agentTraceStatus&gt;</entry>
  <entry key="DestinationTransferStates"/>
  <entry key="queueManager">MFTHAQM</entry>
  <entry key="agentProductVersion">9.1.4.0</entry>
  <entry key="AgentStatusPublishRate">300</entry>
  <entry key="maxSourceTransfers">25</entry>
  <entry key="AgentStatus">STARTED</entry>
  <entry key="maxDestinationTransfers">25</entry>
  <entry key="agentName">SRC</entry>
  <entry key="CommandTimeUTC">2019-05-22T06:02:50Z</entry>
  <entry key="queueManagerChannel">MFT_HA_CHN</entry>
  <entry key="agentInterfaceVersion">6.00</entry>
  <entry key="agentVersion">p914-L191119</entry>
</properties>
```

相关参考

fteCreateAgent

代理程序 GET

IBM MQ Internet Pass-Thru 配置参考信息

IBM MQ Internet Pass-Thru (MQIPT) 使用名为 mqipt.conf 的配置文件来定义路由和控制 MQIPT 服务器的操作。 从 IBM MQ 9.2 开始，还可以在属性文件中指定 **mqiptAdmin** 命令的配置属性。

MQIPT 配置文件

MQIPT 配置文件由多个部分组成。 对于已定义的通过 MQIPT 的每个路由，都有一个 [global] 部分和一个额外的 [route] 部分。

每个节都包含名称/值属性对。 有些属性可能只出现在 [global] 节中，有些属性可能只出现在 [route] 节中，而有些属性可能同时出现在 [route] 和 [global] 节中。 如果某个属性同时出现在路由和 [global] 部分中，那么 [route] 部分中该属性的值将覆盖全局值，但仅针对所讨论的路由。 通过这种方式，可以使用 [global] 部分来确定要用于未在各个 [route] 部分中设置的属性的缺省值。

[global] 节以包含字符串 [global] 的行开头，并结束于第一个 [route] 节的开始处。 [global] 节必须位于文件中的所有 [route] 节前面。

每个 [route] 节都以包含字符串 [route] 的行开头，并结束于下一个 [route] 节的开始处，或结束于配置文件的结尾处。

将忽略任何无法识别的属性名称。 如果 [route] 节中的某个属性具有可识别名称但具有无效值（例如 MinConnectionThreads=x 或 HTTP=unsure），那么将会禁用该路径（即，不会侦听任何入局连接）。

注意: 可以在 mqipt.conf 文件中添加的路由数的最大限制为 100。

[global] 部分中的属性值无效可能会阻止 MQIPT 或命令服务器启动。 如果命令服务器未启动，那么 MQIPT 不会侦听 **mqiptAdmin** 命令发送到受影响命令端口的管理命令。 如果刷新 MQIPT 时存在 [global] 部分中具有无效值的属性，那么将发出一条警告消息，并且该属性的有效值保持不变。 这将防止无效属性值在刷新时导致 MQIPT 的活动实例关闭。

如果属性列示为采用值 true 或 false，那么可以在属性值中同时使用大写和小写字符。

您可以通过编辑 mqipt.conf 文件来更改属性的值。 要应用任何更改，请使用带有 **-refresh** 关键字的 **mqiptAdmin** 命令来刷新 MQIPT 。

要在配置文件中包含注释，请以 " #" 字符开头一行。

仅在已启用其他属性时，更改某些属性才会导致重新启动路径。 例如，仅在已启用 **HTTP** 属性时，更改 HTTP 属性才会生效。

重新启动路径时，会终止现有连接。 要覆盖此行为，请将 **RouteRestart** 属性设置为 false。 这将阻止路由重新启动，从而允许现有连接保持活动状态，直到重新启用 **RouteRestart** 属性为止。

有关如何设置一些简单配置的信息，请参阅 MQIPT 入门。 有关样本配置，请参阅 MQIPT 安装目录中的 mqiptSample.conf 文件。

mqiptAdmin 属性文件

可以在单独的属性文件中指定 **mqiptAdmin** 命令的配置属性。 当 **mqiptAdmin** 连接到 MQIPT TLS 命令端口时，需要这些配置属性。

有关可以在 **mqiptAdmin** 属性文件中指定的属性的列表，请参阅 第 223 页的『mqiptAdmin 属性』。 属性名是区分大小写的。 将忽略任何无法识别的属性。

可以通过以 " #" 字符开头的行在属性文件中包含注释。

MQIPT 属性的摘要

此表显示了 MQIPT 配置属性的摘要并包括以下信息:

- 按字母顺序排列的 MQIPT 属性列表，带有指向 [route] 部分或 [global] 部分（如果 [route] 部分不适用）中的详细信息的链接。
- 必须设置为 true 才能使值生效的属性。
- 属性适用于 [global] 部分还是 [route] 部分，还是同时适用于两者。
- [route] 部分和 [global] 部分中都缺失属性时将使用的缺省值。 在指定值 true 和 false 时，可以使用大写和小写字符的任意组合。

属性名称	要设置为 true 的属性	Global	Route	缺省值
AccessPW		yes	False	空值
Active		yes	yes	true
V 9.4.0 V 9.4.0 第 208 页的『[MQ 9.4.0 2024 年 6 月][MQ 9.4.0 2024 年 6 月]AllowedProtocols』		yes	yes	mq
ClientAccess		yes	yes	false
CommandPort		yes	False	空值
CommandPortListenerAddress		yes	False	空值
ConnectionLog		yes	False	true

属性名称	要设置为 true 的属性	Global	Route	缺省值
Destination		False	yes	空值
DestinationPort		False	yes	1414
第 206 页的『EnableAdvancedCapabilities』		yes	False	false
HTTP		yes	yes	false
V 9.4.0 V 9.4.0 第 209 页的『[MQ 9.4.0 2024 年 6 月][MQ 9.4.0 2024 年 6 月]HTTPConnectionTimeout』		yes	yes	5000
HTTPProxy	HTTP	yes	yes	空值
HTTPProxyPort	HTTP	yes	yes	8080
HTTPS	HTTP	yes	yes	false
HTTPServer	HTTP	yes	yes	空值
HTTPServerPort	HTTP	yes	yes	空值
IdleTimeout		yes	yes	0
IgnoreExpiredCRLs		yes	yes	false
LDAP		yes	yes	false
LDAPIgnoreErrors	LDAP	yes	yes	false
LDAPCacheTimeout	LDAP	yes	yes	24
LDAPServer1	LDAP	yes	yes	空值
LDAPServer1Port	LDAP	yes	yes	389
LDAPServer1Userid	LDAP	yes	yes	空值
LDAPServer1Password	LDAP	yes	yes	空值
LDAPServer1Timeout	LDAP	yes	yes	0
LDAPServer2	LDAP	yes	yes	空值
LDAPServer2Port	LDAP	yes	yes	389
LDAPServer2Userid	LDAP	yes	yes	空值
LDAPServer2Password	LDAP	yes	yes	空值
LDAPServer2Timeout	LDAP	yes	yes	0
ListenerAddress		yes	yes	空值
ListenerPort		False	yes	空值
LocalAddress		yes	yes	空值
LocalAdmin		yes	False	true
MaxConnectionThreads		yes	yes	100
MaxLogFileSize		yes	False	50
MinConnectionThreads		yes	yes	5

属性名称	要设置为 true 的属性	Global	Route	缺省值
Name		False	yes	空值
OutgoingPort		False	yes	0
V 9.4.0 PasswordProtection		yes	yes	Required
QMgrAccess		yes	yes	true
RemoteCommand 认证		yes	False	NONE
RemoteShutdown		yes	False	false
RouteRestart		yes	yes	true
SecurityExit		yes	yes	false
SecurityExitName	SecurityExit	yes	yes	空值
SecurityExitPath	SecurityExit	yes	yes	*mqipt_home*\exits
SecurityExitTimeout	SecurityExit	yes	yes	30
SecurityManager (注 3)		yes	False	false
SecurityManager 策略 (注 3)		yes	False	空值
SocksClient		yes	yes	false
SocksProxyHost	SocksClient	yes	yes	空值
SocksProxyPort	SocksClient	yes	yes	1080
SocksServer		yes	yes	false
SSLClient		yes	yes	false
SSLClientCAKeyRing	SSLClient	yes	yes	空值
SSLClientCAKeyRingPW	SSLClient	yes	yes	空值
第 213 页的『SSLClientCAKeyRingUseCryptoHardware』	SSLClient	yes	yes	false
SSLClientCipherSuites	SSLClient	yes	yes	空值
SSLClientConnectTimeout	SSLClient	yes	yes	30
SSLClientCustomOutboundSNI	SSLClient	yes	yes	空值
SSLClientDN_C	SSLClient	yes	yes	*（注释 1）
SSLClientDN_CN	SSLClient	yes	yes	*（注释 1）
SSLClientDN_DC	SSLClient	yes	yes	*（注释 1）
SSLClientDN_DNQ	SSLClient	yes	yes	*（注释 1）
SSLClientDN_L	SSLClient	yes	yes	*（注释 1）
SSLClientDN_O	SSLClient	yes	yes	*（注释 1）
SSLClientDN_OU	SSLClient	yes	yes	*（注释 1）
SSLClientDN_PC	SSLClient	yes	yes	*（注释 1）

属性名称	要设置为 true 的属性	Global	Route	缺省值
SSLClientDN_ST	SSLClient	yes	yes	*（注释 1）
SSLClientDN_Street	SSLClient	yes	yes	*（注释 1）
SSLClientDN_T	SSLClient	yes	yes	*（注释 1）
SSLClientDN_UID	SSLClient	yes	yes	*（注释 1）
SSLClientExit		yes	yes	false
SSLClientKeyRing	SSLClient	yes	yes	空值
SSLClientKeyRingPW	SSLClient	yes	yes	空值
第 215 页的『SSLClientKeyRingUseCryptoHardware』	SSLClient	yes	yes	false
第 216 页的『SSLClientOutboundSNI』	SSLClient	yes	yes	HOSTNAME
SSLClientProtocols	SSLClient	yes	yes	TLSv1.2 TLSv1.3
SSLClientSiteDN_C	SSLClient	yes	yes	*（注释 1）
SSLClientSiteDN_CN	SSLClient	yes	yes	*（注释 1）
SSLClientSiteDN_DC	SSLClient	yes	yes	*（注释 1）
SSLClientSiteDN_DNQ	SSLClient	yes	yes	*（注释 1）
SSLClientSiteDN_L	SSLClient	yes	yes	*（注释 1）
SSLClientSiteDN_O	SSLClient	yes	yes	*（注释 1）
SSLClientSiteDN_OU	SSLClient	yes	yes	*（注释 1）
SSLClientSiteDN_PC	SSLClient	yes	yes	*（注释 1）
SSLClientSiteDN_ST	SSLClient	yes	yes	*（注释 1）
SSLClientSiteDN_Street	SSLClient	yes	yes	*（注释 1）
SSLClientSiteDN_T	SSLClient	yes	yes	*（注释 1）
SSLClientSiteDN_UID	SSLClient	yes	yes	*（注释 1）
SSLClientSiteLabel	SSLClient	yes	yes	空值
SSLCommandPort		yes	False	空值
SSLCommandPortCipherSuites		yes	False	空值
SSLCommandPortListenerAddress		yes	False	空值
SSLCommandPortKeyRing		yes	False	空值
SSLCommandPortKeyRingPW		yes	False	空值
SSLCommandPortKeyRingUseCrypto 硬件		yes	False	false
SSLCommandPort 协议		yes	False	TLSv1.2 TLSv1.3
SSLCommandPortSiteLabel		yes	False	空值

属性名称	要设置为 **true** 的属性	Global	Route	缺省值
SSLExitData	SSLServerExit	yes	yes	空值
SSLExitName	SSLServerExit	yes	yes	空值
SSLExitPath	SSLServerExit	yes	yes	*mqipt_home* \ exits
SSLExitTimeout	SSLServerExit	yes	yes	30
SSLProxyMode		yes	yes	false
SSLPlainConnections	SSLServer 或 SSLProxyMode	yes	yes	false
SSLServer		yes	yes	false
SSLServerAskClientAuth	SSLServer	yes	yes	false
SSLServerCAKeyRing	SSLServer	yes	yes	空值
SSLServerCAKeyRingPW	SSLServer	yes	yes	空值
第 219 页的『SSLServerCAKeyRingUseCryptoHardware』	SSLServer	yes	yes	false
SSLServerCipherSuites	SSLServer	yes	yes	空值
SSLServerDN_C	SSLServer	yes	yes	*（注释 1）
SSLServerDN_CN	SSLServer	yes	yes	*（注释 1）
SSLServerDN_DC	SSLServer	yes	yes	*（注释 1）
SSLServerDN_DNQ	SSLServer	yes	yes	*（注释 1）
SSLServerDN_L	SSLServer	yes	yes	*（注释 1）
SSLServerDN_O	SSLServer	yes	yes	*（注释 1）
SSLServerDN_OU	SSLServer	yes	yes	*（注释 1）
SSLServerDN_PC	SSLServer	yes	yes	*（注释 1）
SSLServerDN_ST	SSLServer	yes	yes	*（注释 1）
SSLServerDN_Street	SSLServer	yes	yes	*（注释 1）
SSLServerDN_T	SSLServer	yes	yes	*（注释 1）
SSLServerDN_UID	SSLServer	yes	yes	*（注释 1）
SSLServerExit		yes	yes	false
SSLServerKeyRing	SSLServer	yes	yes	空值
SSLServerKeyRingPW	SSLServer	yes	yes	空值
第 221 页的『SSLServerKeyRingUseCryptoHardware』	SSLServer	yes	yes	false
SSLServerProtocols	SSLServer	yes	yes	TLSv1.2 TLSv1.3
SSLServerSiteDN_C	SSLServer	yes	yes	*（注释 1）

属性名称	要设置为 true 的属性	Global	Route	缺省值
SSLServerSiteDN_CN	SSLServer	yes	yes	*（注释 1）
SSLServerSiteDN_DC	SSLServer	yes	yes	*（注释 1）
SSLServerSiteDN_DNQ	SSLServer	yes	yes	*（注释 1）
SSLServerSiteDN_L	SSLServer	yes	yes	*（注释 1）
SSLServerSiteDN_O	SSLServer	yes	yes	*（注释 1）
SSLServerSiteDN_OU	SSLServer	yes	yes	*（注释 1）
SSLServerSiteDN_PC	SSLServer	yes	yes	*（注释 1）
SSLServerSiteDN_ST	SSLServer	yes	yes	*（注释 1）
SSLServerSiteDN_Street	SSLServer	yes	yes	*（注释 1）
SSLServerSiteDN_T	SSLServer	yes	yes	*（注释 1）
SSLServerSiteDN_UID	SSLServer	yes	yes	*（注释 1）
SSLServerSiteLabel	SSLServer	yes	yes	空值
StoredCredentials 格式		yes	yes	空值
TCPKeepAlive		yes	yes	false
Trace		yes	yes	0
V 9.4.0 traceFileCount		yes	False	25
V 9.4.0 traceFileSize		yes	False	200
第 223 页的『TraceUserData』		yes	yes	64
UriName	HTTP	yes	yes	（注释 2）

注意:

1. 星号 (*) 表示通配符。
2. 请参阅第 208 页的『MQIPT 路由属性』中的 UriName 获取有关缺省设置的详细信息。
3. Deprecated 不推荐在将来的发行版中除去此属性。

相关参考

第 199 页的『IBM MQ Internet Pass-Thru 配置参考信息』
IBM MQ Internet Pass-Thru (MQIPT) 使用名为 mqipt.conf 的配置文件来定义路由和控制 MQIPT 服务器的操作。 从 IBM MQ 9.2 开始，还可以在属性文件中指定 **mqiptAdmin** 命令的配置属性。

第 205 页的『MQIPT 全局属性』
mqipt.conf 配置文件可以包含许多全局属性。

第 208 页的『MQIPT 路由属性』
mqipt.conf 配置文件可以包含各个路由的属性。

MQIPT 全局属性

mqipt.conf 配置文件可以包含许多全局属性。

以下属性只能出现在 mqipt.conf 的 [global] 节中。 除 **ListenerPort**， **Destination**，**DestinationPort**， **Name** 和 **OutgoingPort** 以外的所有 路由属性 也可以显示在 [global] 部分中。 如果某个属性同时出现在路由和 [global] 部分中，那么 [route] 部分中该属性的值将覆盖全局值，但仅

针对所讨论的路由。 通过这种方式，可以使用 [global] 部分来确定要用于未在各个 [route] 部分中设置的属性的缺省值。

AccessPW
用于认证使用 **mqiptAdmin** 命令发送到 MQIPT 命令端口的命令的密码。

该值可以是已使用 **mqiptPW** 命令加密的密码，也可以是纯文本密码。 纯文本密码只能包含字母数字字符。 强烈建议您对 MQIPT 配置中存储的密码进行加密。 有关在 MQIPT 配置中加密密码的更多信息，请参阅 加密存储的密码。

如果满足以下两个条件，那么将对命令端口接收的管理命令执行认证:

- 指定了 **AccessPW** 属性并将其设置为非空值。
- **RemoteCommandAuthentication** 属性已指定并设置为除 none 以外的值。

CommandPort
不受保护的命令端口的 TCP/IP 端口号。 MQIPT 接受 **mqiptAdmin** 命令发送到此命令端口的管理命令。

与不受保护的命令端口的连接不受 TLS 保护。 发送到命令端口的数据 (包括访问密码) 可能由网络的其他用户访问。 要配置使用 TLS 保护的命令端口，请改为设置 **SSLCommandPort** 属性。

如果未指定 **CommandPort** 属性，那么 MQIPT 不会侦听不受保护的命令端口上的管理命令。 要使用 **mqiptAdmin** 命令缺省使用的缺省端口号 1881，请将 **CommandPort** 设置为 1881。

CommandPortListenerAddress
要由不受保护的命令端口使用的本地侦听器地址。 通过设置本地侦听器地址，可以将非安全命令端口的入站连接限制为来自特定网络接口的入站连接。 缺省值是侦听所有网络接口。

ConnectionLog
true 或 false。 如果此属性为 true，那么 MQIPT 将会在 logs 子目录中记录所有连接尝试（无论是否成功），并 将断开连接事件记录到 mqipt*YYYYMMDDHHmmSS*.log 文件（其中 *YYYYMMDDHHmmSS* 是表示当前日期和时间的字符）。 **ConnectionLog** 的缺省值为 true。 当此属性从 true 更改为 false 时，MQIPT 将会关闭现有连接日志并创建一个新日志。 当此属性重置为 true 时，将使用新日志。

EnableAdvancedCapabilities
将此属性设置为 true 以确认需要 IBM MQ Advanced， IBM MQ Advanced for z/OS， IBM MQ Advanced for z/OS VUE， 或 IBM MQ Appliance 权利的高级功能可以由 MQIPT 使用。 如果您具有相应的权利，那么可以使用 MQIPT 中的高级功能。 如果在路由上启用了高级功能，那么还需要使用 MQIPT 路由连接的本地队列管理器具有 IBM MQ Advanced， IBM MQ Advanced for z/OS， IBM MQ Advanced for z/OS VUE， 或 IBM MQ Appliance 权利。 除非此属性设置为 true，否则无法启动使用高级功能的路由。 当此属性从 true 更改为 false 时，将停止使用高级功能的路由。

LocalAdmin
指定是否允许没有命令端口的本地管理。 如果此属性设置为 false，那么不接受 **mqiptAdmin** 命令使用本地管理而不是命令端口发送的管理命令。

此属性的有效值为 true 和 false。 缺省值为 true。

MaxLogFileSize
连接日志文件的最大大小（以 KB 为单位来指定）。 当文件大小增加到超过此最大值时，将会生成备份副本 (mqipt001.log)，并启动新文件。 仅保留两个备份文件（ mqipt001.log 和 mqipt002.log）； 每次主日志文件写满时，都会擦除较早的备份。 **MaxLogFileSize** 的缺省值为 50；允许的最小值为 5。

RemoteCommandAuthentication
指定是否应认证不受保护的命令端口或 TLS 命令端口接收的管理命令。 通过检查提供的密码是否与 AccessPW 属性中指定的密码匹配来认证命令。 该值可以是下列任一值:

none
不会对向任一命令端口发出的命令执行认证。 **mqiptAdmin** 命令的用户无需输入密码。 这是缺省值。

可选
mqiptAdmin 命令的用户无需提供密码。 但是，如果提供了密码，那么该密码必须有效。

必需
mqiptAdmin 命令的用户需要提供对命令端口发出的每个命令的有效密码。

还必须指定 **AccessPW** 属性以对命令端口启用认证。

RemoteShutDown
指定是否可以通过 **mqiptAdmin** 命令发送到不受保护的命令端口或 TLS 命令端口的停止命令来关闭 MQIPT。 对于要处理的任一命令端口接收的停止命令，必须将此属性设置为 `true`。

此属性的有效值为 `true` 和 `false`。 缺省值为 `false`。

SecurityManager
将此属性设置为 `true` 以对此 MQIPT 实例启用 Java security manager。 您必须确保已获得正确的许可权。 请参阅 Java security manager 以获取更多信息。 此属性的缺省值为 `false`。

不推荐在将来的发行版中除去此属性。

SecurityManagerPolicy
Java security manager 策略文件的标准文件名。 如果未设置此属性，那么仅使用缺省系统和用户策略文件。 如果已启用 Java security manager，那么在禁用并重新启用 Java security manager 之后，对此属性的更改才会生效。

Deprecated 不推荐在将来的发行版中除去此属性。

SSLCommandPort
TLS 命令端口的 TCP/IP 端口号。 MQIPT 接受 **mqiptAdmin** 命令发送到此命令端口的管理命令。 此端口仅接受 TLS 连接。 必须指定此属性才能启用 TLS 命令端口。

SSLCommandPortCipherSuites
要在 TLS 命令端口上启用的密码套件的名称。 可以通过使用逗号分隔值来指定多个密码套件。 只能指定与 MQIPT 一起提供的 Java runtime environment (JRE) 中默认启用的 TLS 1.2 和 TLS 1.3 密码套件。 如果未指定此属性，则在 JRE 中启用的所有密码套件都会在 TLS 命令端口上启用。

SSLCommandPortListenerAddress
要由 TLS 命令端口使用的本地侦听器地址。 通过设置本地侦听器地址，可以将 TLS 命令端口的入站连接限制为来自特定网络接口的入站连接。 缺省值是侦听所有网络接口。

SSLCommandPortKeyRing
包含 TLS 命令端口服务器证书的 PKCS#12 密钥环文件的名称。

在 Windows 平台，您必须使用双反斜杠 (\\) 作为文件分隔符。

SSLCommandPortKeyRingPW
用于访问 TLS 命令端口密钥环文件或 PKCS #11 密钥库的加密密码。 必须使用 `mqiptPW` 命令对密码进行加密，并将此属性的值设置为 `mqiptPW` 输出的字符串。

SSLCommandPortKeyRingUseCryptoHardware
指定是否将支持 PKCS #11 接口的加密硬件用作 TLS 命令端口服务器证书的密钥库。 此属性的有效值为 `true` 和 `false`。 如果此属性设置为 `true`，那么也不能指定 **SSLCommandPortKeyRing**。

在 MQIPT 中使用加密硬件是一项 IBM MQ Advanced 功能。 `EnableAdvancedCapabilities` 属性必须设置为 `true` 以确认您具有 IBM MQ Advanced 权利。

SSLCommandPortProtocols
要在 TLS 命令端口上启用的协议的逗号分隔列表。 可以指定以下一个或多个值。

表 *117:* 命令端口 *TLS* 协议的允许值

值	**协议**
`TLSv1.2`	TLS 1.2
`TLSv1.3`	TLS 1.3

如果未指定此属性，那么缺省情况下将启用 TLS 1.2 和 TLS 1.3。

SSLCommandPortSiteLabel
TLS 命令端口使用的服务器证书的标签名称。 如果未指定此属性，那么将选择 TLS 命令端口密钥库中与密码套件兼容的任何证书。

跟踪

与路径无关的全局 MQIPT 线程以及未设置 **Trace** 属性的路径的跟踪级别。 例如，主 MQIPT 控制线程和命令服务器线程不与路由关联，仅当在 `[global]` 部分中启用了跟踪时才会进行跟踪。 `[route]` 节

中 **Trace** 属性的值将覆盖该路径的全局 **Trace** 属性。 有关与路由关联的跟踪线程的信息，请参阅 [route] 部分中的 **Trace**。

此属性的值可以是下列其中一项:

0
未启用跟踪

任何正整数
已启用跟踪

缺省值为 0。

V 9.4.0 **TraceFileCount**

MQIPT 用于写入跟踪数据的旋转文件集中的跟踪文件数。

允许的最小值为 3。 缺省值为 25。

如果更改此属性的值，那么将关闭当前跟踪文件，并打开循环跟踪文件集中的下一个文件。

V 9.4.0 **TraceFileSize**

MQIPT 生成的跟踪文件的最大大小，以 MB 为单位指定。

允许的最小值为 1。 缺省值为 200。

如果更改此属性的值，那么将关闭当前跟踪文件，并打开循环跟踪文件集中的下一个文件。

MQIPT 路由属性

mqipt.conf 配置文件可以包含各个路由的属性。

mqipt.conf 配置文件的 [route] 部分可包含以下属性:

活动
仅在 **Active** 的值设置为 true 时，路由接受入站连接。 这意味着，您可以临时关闭对目标的访问，方法是将此值设置为 false，而不需要删除配置文件中的 [route] 部分。 如果将此属性更改为 false，那么发出刷新命令时路由停止。 路由的所有连接停止。

V 9.4.0 V 9.4.0 **AllowedProtocols**
指定此路由接受的协议。 使用此列表中未指定的协议的连接将被拒绝。 可以将多个协议指定为逗号分隔列表。 可以指定以下协议:

mq
该路由接受使用 IBM MQ 协议的连接。

http
该路由接受来自另一个 MQIPT 实例的 HTTP 连接。

此属性的缺省值为 mq。

如果此属性的值已更改，那么路由将停止，并在发出刷新命令时重新启动。 路由的所有连接停止。

ClientAccess
仅在 **ClientAccess** 的值设置为 true 时，路由允许入站客户机通道连接。 请注意，您可以将 MQIPT 配置为仅接受客户机请求或队列管理器请求，或者两种请求都接受。 此属性与 **QMgrAccess** 属性一起使用。 如果将此属性更改为 false，那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

Destination
此路由要连接到的队列管理器或后续 MQIPT 实例的主机名 (或点分十进制 IP 地址)。 每一个 [route] 部分必须包含明确的 **Destination** 值，但是几个 [route] 部分可以引用相同的目标。 如果此属性的更改影响路由，那么路由停止，并且发出刷新命令时重新启动。 路由的所有连接停止。 使用 **SocksProxyHost** 属性时， **Destination** 属性必须使用点分十进制 IPv4 地址格式。

DestinationPort
此路由要连接的目标主机上的端口。 每一个 [route] 部分必须包含明确的 **DestinationPort** 值，但是几个路由可以引用相同组合的 **Destination** 和 **DestinationPort** 值。 如果此属性的更改影响路由，那么路由停止，并且发出刷新命令时重新启动。 路由的所有连接停止。

HTTP
针对负责发出出站 HTTP 隧道传送请求的路由，将 **HTTP** 设置为 true。 当路由的 **Destination** 属性 **HTTP** 设置为 true 时，该属性必须是另一个 MQIPT 实例的主机名。 对于连接到 IBM MQ 队列管理器的路由，将 **HTTP** 设置为 false 。 如果更改此属性，将停止路由。 当 **HTTP** 设置为 true 时，还必须至少指定其中一个 **HTTPProxy** 或 **HTTPServer** 属性。 此属性不能与 **SocksClient** 属性一起使用。

V 9.4.0 V 9.4.0 **HTTPConnectionTimeout**
此属性指定在拒绝连接之前， MQIPT 等待接收到的 HTTP 连接成功建立的时间 (以毫秒为单位)。

缺省值是 5000。

HTTPProxy
此路由的所有连接使用的 HTTP 代理的主机名（或点分十进制 IP 地址）。 将向 HTTP 代理发出 **CONNECT** 请求，而不是在未配置 HTTP 代理时通常使用的 **POST** 请求。 如果更改此属性（并且 **HTTP** 设置为 true），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

HTTPProxyPort
HTTP 代理上使用的端口地址。 缺省值为 8080。 如果更改此属性（并且 **HTTP** 设置为 true），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

HTTPServer
此路由的所有连接使用的 HTTP 服务器的主机名（或点分十进制 IP 地址）。 这通常是另一个 MQIPT 的主机名。

如果未指定 **HTTPProxy** ，那么 MQIPT 将连接到 **HTTPServer** 中指定的主机，并向路由 **Destination** 属性中指定的主机发出 HTTP **POST** 请求。 如果指定了 **HTTPProxy** ，那么 MQIPT 将改为连接到 **HTTPProxy** 中指定的主机，并请求代理建立与 **HTTPServer** 中指定的主机的隧道。

如果指定了 **HTTPProxy** ，那么缺省值为路径 **Destination**。

如果更改此属性（并且 **HTTP** 设置为 true），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

HTTPS
将 **HTTPS** 设置为 true 以发出 HTTPS 请求。 对于 SSL/TLS 操作，还必须启用 **HTTP** 和 **SSLClient** 属性，并使用 **SSLClientKeyRing** 或 **SSLClientKeyRingUseCryptoHardware** 属性配置客户机密钥环。 如果更改 **HTTPS** 属性（并且 **HTTP** 设置为 true），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

HTTPServerPort
HTTP Server 上使用的端口地址。 除非指定了 **HTTPProxy** ，否则缺省值为 8080，在这种情况下，缺省值为路径 **DestinationPort**。

如果更改此属性（并且 **HTTP** 设置为 true），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

IdleTimeout
空闲连接经过该时长（分钟）后关闭。 请注意，队列管理器通道的队列管理器也具有 **DISCINT** 属性。 如果设置 **IdleTimeout** 属性，请注意 **DISCINT**。 如果 **IdleTimeout** 设置为 0，没有空闲超时。 对此属性的更改仅在重新启动路由时生效。

IgnoreExpiredCRLs
将 **IgnoreExpiredCRLs** 设置为 true 以忽略到期的 CRL。 缺省值为 false。 请注意，如果将 **IgnoreExpiredCRLs** 设置为 true，那么可以使用已撤销的证书来建立 SSL/TLS 连接。

LDAP
将 **LDAP** 设置为 true 以允许在使用 SSL/TLS 连接时使用 LDAP 服务器。 MQIPT 将使用 LDAP 服务器检索 CRL 和 ARL。 要让此属性生效，还必须将 **SSLClient** 属性或 **SSLServer** 属性设置为 true。

LDAPCacheTimeout
存储从 LDAP 服务器检索到的 CRL 的临时缓存的到期时间（小时）。 经过此时间之后，整个 CRL 缓存被清空。 例如，指定值为 1 小时表明缓存每小时清空一次。 缺省值为 24。 如果指定超时值为 0，那么直

到重新启动路由后缓存中的条目才到期。如果更改此属性 (并且 **LDAP** 设置为 true)，那么路由将停止，并在发出刷新命令时重新启动。 路由的所有连接停止。

LDAPIgnoreErrors
将 **LDAPIgnoreErrors** 设置为 true 以在执行 LDAP 搜索时忽略任何连接或超时错误。 如果 MQIPT 不能成功执行搜索，那么将不允许客户机连接完成，除非已启用此属性。 成功搜索意味着已经检索 CRL，或者指定的 CA 没有可用的 CRL。 如果更改此属性 (并且 **LDAP** 设置为 true)，那么路由将停止，并在发出刷新命令时重新启动。 路由的所有连接停止。

注: 如果启用此属性，那么可使用已撤销的证书来建立 SSL/TLS 连接。

LDAPServer1
主要 LDAP 服务器的主机名或 IP 地址。 如果 LDAP 已经设置为 true，必须设置此属性。如果更改此属性 (并且 **LDAP** 设置为 true)，那么路由将停止，并在发出刷新命令时重新启动。 路由的所有连接停止。

LDAPServer1Port
主要 LDAP 服务器的侦听端口号。 缺省值为 389。如果更改此属性 (并且 **LDAP** 设置为 true)，那么路由将停止，并在发出刷新命令时重新启动。 路由的所有连接停止。

LDAPServer1Userid
访问主要 LDAP 服务器所需的用户标识。 如果需要访问主要 LDAP 服务器的权限，那么必须设置此属性。如果更改此属性 (并且 **LDAP** 设置为 true)，那么路由将停止，并在发出刷新命令时重新启动。 路由的所有连接停止。

LDAPServer1Password
访问主要 LDAP 服务器所需的密码。 如果 **LDAPServer1Userid** 已经设置为 true，必须设置此属性。如果更改此属性 (并且 **LDAP** 设置为 true)，那么路由将停止，并在发出刷新命令时重新启动。 路由的所有连接停止。

该值可以是已使用 **mqiptPW** 命令加密的密码，也可以是纯文本密码。 纯文本密码只能包含字母数字字符。 强烈建议您对 MQIPT 配置中存储的密码进行加密。 有关在 MQIPT 配置中加密密码的更多信息，请参阅 加密存储的密码。

LDAPServer1Timeout
MQIPT 等待主要 LDAP 服务器响应的时间（秒）。 缺省值为 0，这意味着连接将不超时。如果更改此属性 (并且 **LDAP** 设置为 true)，那么路由将停止，并在发出刷新命令时重新启动。 路由的所有连接停止。

LDAPServer2
备份 LDAP 服务器的主机名或 IP 地址。 此属性为可选属性。如果更改此属性 (并且 **LDAP** 设置为 true)，那么路由将停止，并在发出刷新命令时重新启动。 路由的所有连接停止。

LDAPServer2Port
备份 LDAP 服务器的侦听端口号。 缺省值为 389。如果更改此属性 (并且 **LDAP** 设置为 true)，那么路由将停止，并在发出刷新命令时重新启动。 路由的所有连接停止。

LDAPServer2Userid
访问备份 LDAP 服务器所需的用户标识。 如果需要访问备份 LDAP 服务器的权限，那么必须设置此属性。如果更改此属性 (并且 **LDAP** 设置为 true)，那么路由将停止，并在发出刷新命令时重新启动。 路由的所有连接停止。

LDAPServer2Password
访问备份 LDAP 服务器所需的密码。 如果 **LDAPServer2** 已经设置为 true，必须设置此属性。如果更改此属性 (并且 **LDAP** 设置为 true)，那么路由将停止，并在发出刷新命令时重新启动。 路由的所有连接停止。

该值可以是已使用 **mqiptPW** 命令加密的密码，也可以是纯文本密码。 纯文本密码只能包含字母数字字符。 强烈建议您对 MQIPT 配置中存储的密码进行加密。 有关在 MQIPT 配置中加密密码的更多信息，请参阅 加密存储的密码。

LDAPServer2Timeout
MQIPT 等待备份 LDAP 服务器响应的时间（秒）。 缺省值为 0，这意味着连接将不超时。如果更改此属性 (并且 **LDAP** 设置为 true)，那么路由将停止，并在发出刷新命令时重新启动。 路由的所有连接停止。

ListenerAddress
如果 MQIPT 系统具有多个 IP 地址并且您需要将路由侦听器端口绑定到特定地址，使用此属性。 这可用于将入站连接限制在特定网络接口的地址。 此属性值表示的 IP 地址应属于运行 MQIPT 的系统上的一个网络接口。 缺省情况下接受所有网络接口的连接。

ListenerPort
路由侦听入站请求的端口号。 每个 [route] 部分必须包含明确的 **ListenerPort** 值。在每一个部分设置的 **ListenerPort** 值必须唯一。 可以使用任何有效的端口号，包括端口 80 和 443，前提是同一主机上运行的任何其他 TCP/IP 侦听器没有使用选择的端口。

LocalAddress
此计算机上此路由的所有连接绑定到的 IP 地址。 所选的地址必须是与运行 MQIPT 的计算机上的某一个网络接口关联的 IP 地址。 如果更改此属性，那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

MaxConnectionThreads
此路由可以处理的最大连接线程数，也可以是最大并发连接数。 如果达到此限制，**MaxConnectionThreads** 值还指示所有线程都在使用中时排队的连接数。 超过此数量之后，随后的连接请求将被拒绝。

允许的最小值是 1 和 **MinConnectionThreads** 值中的最大值。

如果增大该值，那么发出刷新命令时将使用新值。 所有连接立即使用新值。 路由不停止。

如果值减小，那么新值仅在路由重新启动时生效。

MinConnectionThreads
启动路由时，分配的用于处理路由上入站连接的连接线程数。 路由处于活动状态期间，分配的线程数不低于此值。

该值必须在范围 0 到 **MaxConnectionThreads** 的值之间。

对此属性的更改仅在重新启动路由时生效。

名称
帮助识别路由的名称。 此属性为可选属性。 此值在控制台消息和跟踪信息中显示。 对此属性的更改仅在重新启动路由时生效。

OutgoingPort
出站连接使用的起始端口号。 端口号范围与此路由的 **MaxConnectionThread** 值匹配。 缺省值 0 使用系统定义的端口号。 如果更改此属性，那么路由停止并在发出刷新命令时重新启动。 此路由的所有连接停止。 使用 HTTP 时，每一个通道连接都需要两个出站端口。 有关更多信息，请参阅 端口号控制。

V 9.4.0 **PasswordProtection**
指定对于配置为添加或除去 TLS 加密的 MQIPT 路由， MQIPT 是否可以为 IBM MQ clients 在 MQCSP 结构中发送的凭证添加或除去保护，以保持客户机与队列管理器之间的兼容性。

MQCSP 结构中的凭证可以使用 IBM MQ MQCSP 密码保护功能进行保护，也可以使用 TLS 加密进行加密。 MQCSP 密码保护对于测试和开发目的很有用，因为它比设置 TLS 加密更简单，但它没有那么安全。

有关 MQCSP 密码保护的更多信息，请参阅 MQCSP 密码保护。

当 MQIPT 路由配置为添加或移除 TLS 加密时， MQIPT 可能需要保护 MQCSP 结构中的凭证，或者移除 MQCSP 密码保护，才能成功连接。

该属性的值可以是下列其中一个值:

必需
MQIPT 确保 MQCSP 结构中的凭证通过使用 TLS 进行加密或通过 MQCSP 密码保护进行保护。

如果客户机使用 TLS 加密来发送 MQCSP 结构中的凭证，并且 MQIPT 路由除去 TLS 加密，那么在将凭证转发到路由目标之前， MQIPT 会使用 MQCSP 密码保护来保护凭证。 当使用 SSLServer=true 和 SSLClient=false 配置了 MQIPT 路由，并且所选 CipherSuite 未使用空密码时，会发生此情况。

如果 MQCSP 结构中的凭证受具有 MQCSP 密码保护的客户机保护，那么 MQIPT 不会除去该保护，即使 MQIPT 与路由目标之间的连接使用 TLS 加密也是如此。 如果 MQIPT 与路由目标之间的连接使用 TLS 加密，那么连接可能会失败，原因码为 MQRC_PASSWORD_PROTECTION_ERROR (2594)。

这是缺省值。

兼容

MQIPT 根据需要应用或除去 MQCSP 密码保护，以确保连接成功。

如果客户机使用 TLS 加密来发送 MQCSP 结构中的凭证，并且 MQIPT 路由除去 TLS 加密，那么 MQIPT 会在将密码转发到路由目标之前使用 MQCSP 密码保护来保护凭证。 当使用 SSLServer=true 和 SSLClient=false 配置了 MQIPT 路由，并且所选 CipherSuite 未使用空密码时，会发生此情况。

如果 MQCSP 结构中的凭证受具有 MQCSP 密码保护的客户机保护，并且 MQIPT 路由添加 TLS 加密，那么 MQIPT 会在将凭证转发到路由目标之前除去 MQCSP 密码保护。 当使用 SSLServer=false 和 SSLClient=true 配置了 MQIPT 路由，并且所选 CipherSuite 未使用空密码时，会发生此情况。

此选项提供最佳兼容性。 但是，它只能用于可信网络上的测试和开发目的，因为它不会确保在网络上保护密码。

中继

MQCSP 结构中的凭证由 MQIPT 转发到路由目标，而无需添加或除去 MQCSP 密码保护。 如果 MQIPT 路由配置为添加或除去 TLS 加密，那么客户机连接可能会失败，原因码为 MQRC_PASSWORD_PROTECTION_ERROR (2594)。

QMgrAccess

将 **QMgrAccess** 设置为 `true` 以允许入站队列管理器通道连接（例如，发送方通道）。 如果将此属性更改为 `false`，那么发出刷新命令时路由停止。 此路由的所有连接停止。

RouteRestart

将 **RouteRestart** 设置为 `false` 以在更改其他路由属性并且发出刷新命令时阻止该路由重新启动。 此属性的缺省值为 `true`。

SecurityExit

将 **SecurityExit** 设置为 `true` 以启用用户定义的安全出口。 此属性的缺省值为 `false`。

SecurityExitName

用户定义的安全出口的类名。 如果 **SecurityExit** 已经设置为 `true`，必须设置此属性。 如果更改此属性（并且 **SecurityExit** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 此路由的所有连接停止。

SecurityExitPath

包含用户定义的安全出口的标准路径名。 如果未设置此属性，它将缺省为出口子目录。 此属性还可以定义包含用户定义的安全出口的 Java 归档 (JAR) 文件的名称。 如果更改此属性（并且 **SecurityExit** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 此路由的所有连接停止。

SecurityExitTimeout

MQIPT 用于确定验证连接请求时等待响应的时间的超时值（秒）。 缺省值为 `30`。 如果更改此属性（并且 **SecurityExit** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SocksClient

将 **SocksClient** 设置为 `true` 以使路由充当 SOCKS 客户机并使用 **SocksProxyHost** 和 **SocksProxyPort** 属性定义通过 SOCKS 代理进行所有连接。 如果更改此属性，那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。 此属性不能与以下项一起使用：

- **HTTP**
- **SocksServer**
- **SSLClient**
- **SSLProxyMode**

SocksProxyHost

此路由的所有连接使用的 SOCKS 代理的主机名 (或点分十进制 IPv4 地址)。 如果更改此属性（并且 **SocksClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 此路由的所有连接停止。 使用 **SocksProxyHost** 属性时，**Destination** 属性必须使用点分十进制格式。

SocksProxyPort
SOCKS 代理上使用的端口号。 缺省值为 1080。 如果更改此属性（并且 **SocksClient** 设置为 true），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SocksServer
将 **SocksServer** 设置为 true 以使路由充当 SOCKS 代理并接受 SOCKS 客户机连接。 如果更改此属性，那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。 此属性不能与以下属性一起使用：

- **SocksClient**
- **SSLProxyMode**
- **SSLServer**

SSLClient
将 **SSLClient** 设置为 true 以使路由充当 SSL/TLS 客户机并建立传出 SSL/TLS 连接。 将 **SSLClient** 设置为 true 意味着目标是充当 SSL/TLS 服务器的另一个 MQIPT 实例或 HTTP 代理/服务器。

如果将 **SSLClient** 设置为 true，那么必须使用 **SSLClientKeyRing** 或 **SSLClientCAKeyRing** 属性指定 SSL/TLS 客户机密钥环，或者通过设置 **SSLClientKeyRingUseCryptoHardware** 或 **SSLClientCAKeyRingUseCryptoHardware** 属性将 MQIPT 配置为使用加密硬件。

如果更改 **SSLClient**，那么路由停止并在发出刷新命令时重新启动。 此路由的所有连接停止。

此属性不能与以下属性一起使用：

- **SSLProxyMode**

SSLClientCAKeyRing
包含 CA 证书的密钥环文件的标准文件名，用于认证来自 SSL/TLS 服务器的证书。 在 Windows 平台，您必须使用双反斜杠 (\\) 作为文件分隔符。 如果更改此属性（并且 **SSLClient** 设置为 true），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientCAKeyRingPW
用于打开使用 **SSLClientCAKeyRing** 属性指定的 SSL/TLS 客户机 CA 密钥环文件的密码，或者用于在 **SSLClientCAKeyRingUseCryptoHardware** 属性设置为 true 时连接到加密硬件密钥库的密码。

该值可以是已使用 **mqiptPW** 命令加密的密码，也可以是包含加密密码的文件的标准文件名。 如果在 Windows 平台上指定文件名，那么必须使用双反斜杠 (\\) 作为文件分隔符。 建议您迁移当前存储在文件中的任何密钥环密码，以使用最新且最安全的保护方法，方法是使用 **mqiptPW** 实用程序对密码进行重新加密。 有关在 MQIPT 配置中加密密码的更多信息，请参阅 加密存储的密码。

如果更改此属性（并且 **SSLClient** 设置为 true），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientCAKeyRingUseCryptoHardware

指定当 MQIPT 充当 SSL/TLS 客户机时，是否将支持 PKCS #11 接口的加密硬件用作用于从 SSL/TLS 服务器认证服务器证书的 CA 证书的密钥库。 如果此属性设置为 true，那么不能在同一路径上设置 **SSLClientCAKeyRing** 。

如果更改此属性（并且 **SSLClient** 设置为 true），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

将加密硬件与 MQIPT 配合使用是 IBM MQ Advanced 功能。 要使用此功能，还需要使用 MQIPT 路由连接的本地队列管理器具有 IBM MQ Advanced， IBM MQ Appliance， IBM MQ Advanced for z/OS 或 IBM MQ Advanced for z/OS VUE 权利。 当此属性设置为 true 时，路由将不会启动，除非 **EnableAdvancedCapabilities** 全局属性设置为确认可以使用 IBM MQ Advanced 功能。

SSLClientCipherSuites
要在 SSL/TLS 客户端上使用的 SSL/TLS CipherSuite 的名称。 可以是所支持的一个或多个 CipherSuite。 如果将此属性留空，那么将使用与密钥环中的客户机证书兼容的已启用协议的任何 CipherSuite 。 如果更改此属性（并且 **SSLClient** 设置为 true），那么路由停止并在发出刷新命令时重新启动。 此路由的所有连接停止。

SSLClientConnectTimeout
SSL/TLS 客户机等待接受 SSL/TLS 连接的时间 (以秒计)。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientCustomOutboundSNI
指定当 MQIPT 启动到路由目标的 TLS 连接时 服务器名称指示 (SNI) 的值 (如果将路由配置为 **SSLClientOutboundSNI** 设置为 `custom`)。 使用此属性将 SNI 设置为 MQIPT 无法自动设置的特定值。 例如，如果要将 SNI 设置为主机名，但使用 IP 地址配置了路由目标。

该值必须是符合 RFC 3490 规范的有效国际化域名 (IDN)，并且不能以结尾点结尾。 如果指定了无效值，那么路由不会启动。

如果更改此属性的值，并且 **SSLClientOutboundSNI** 设置为 `custom`，那么发出刷新命令时将停止并重新启动路由。

注意: 在将连接转发到在通道 **CERTLABL** 字段中配置了证书标签的 IBM MQ 通道时，不得使用此设置。 如果以这种方式转发客户机，那么将拒绝该客户机，并在远程队列管理器错误日志中显示 MQRC_SSL_INITIALIZATION_ERROR 返回码和 AMQ9673 错误。

SSLClientDN_C
使用此属性可接受从 SSL/TLS 服务器收到的与此国家或地区名称匹配的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，将接受所有国家或地区名称。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientDN_CN
使用此属性可接受从 SSL/TLS 服务器接收的与此公共名称匹配的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，将接受所有通用名称。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientDN_DC
使用此属性可接受从与此域组件匹配的 SSL/TLS 服务器接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 您可以指定多个 DC，并使用逗号分隔。 每个 DC 表示域名中的一个元素，例如，使用逗号分隔多个值时域名 `example.ibm.com` 表示为 `example,ibm,com`。 如果您不指定此属性，将接受所有域组件。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientDN_DNQ
使用此属性可接受从 SSL/TLS 服务器收到的与此域限定符匹配的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，将接受所有域限定符。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientDN_L
使用此属性可接受从 SSL/TLS 服务器收到的与此位置匹配的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，则暗示“所有位置”。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientDN_O
使用此属性可接受从与此组织匹配的 SSL/TLS 服务器接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，将接受所有组织的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientDN_OU
使用此属性可接受从与此组织单元 (OU) 匹配的 SSL/TLS 服务器接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 您可以指定多个 OU，并使用逗号分隔。 （在文字逗号前加反斜杠 (\) 字符来匹配文字逗号）。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何 OU 名称的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 此路由的所有连接停止。

SSLClientDN_PC
使用此属性可接受从 SSL/TLS 服务器收到的与此邮政编码匹配的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，将接受所有邮政编码。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientDN_ST
使用此属性可接受从 SSL/TLS 服务器收到的与此状态匹配的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，所有州/省的服务器都会接受证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientDN_Street
使用此属性可接受从 SSL/TLS 服务器接收的与此街道名称匹配的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，将接受所有街道名称。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientDN_T
使用此属性可接受从 SSL/TLS 服务器收到的与此标题匹配的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，将接受所有标题。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientDN_UID
使用此属性可接受从 SSL/TLS 服务器收到的与此用户标识匹配的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，将接受所有用户标识。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientExit
使用此属性可在路由充当 SSL/TLS 客户机时启用或禁用出口。 这允许您在配置文件中定义出口详细信息，而不必真正使用它们。

SSLClientKeyRing
包含客户机证书的密钥环文件的标准文件名。 在 Windows 平台，您必须使用双反斜杠 (\\) 作为文件分隔符。 如果更改 **SSLClientKeyRing** （并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientKeyRingPW
用于打开使用 **SSLClientKeyRing** 属性指定的 SSL/TLS 客户机密钥环文件或连接到加密硬件密钥库 (如果 **SSLClientKeyRingUseCryptoHardware** 属性设置为 `true`) 的密码。

该值可以是已使用 **mqiptPW** 命令加密的密码，也可以是包含加密密码的文件的标准文件名。 如果在 Windows 平台上指定文件名，那么必须使用双反斜杠 (\\) 作为文件分隔符。 建议您迁移当前存储在文件中的任何密钥环密码，以使用最新且最安全的保护方法，方法是使用 **mqiptPW** 实用程序对密码进行重新加密。 有关在 MQIPT 配置中加密密码的更多信息，请参阅 加密存储的密码。

如果更改 **SSLClientKeyRingPW** （并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientKeyRingUseCryptoHardware

指定当 MQIPT 充当 SSL/TLS 客户机时，是否将支持 PKCS #11 接口的加密硬件用作包含客户机证书的密钥库。 如果此属性设置为 `true`，那么不能在同一路径上设置 **SSLClientKeyRing** 。

如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

将加密硬件与 MQIPT 配合使用是 IBM MQ Advanced 功能。 要使用此功能，还需要使用 MQIPT 路由连接的本地队列管理器具有 IBM MQ Advanced， IBM MQ Appliance， IBM MQ Advanced for z/OS 或 IBM MQ Advanced for z/OS VUE 权利。 当此属性设置为 `true` 时，路由将不会启动，除非 **EnableAdvancedCapabilities** 全局属性设置为确认可以使用 IBM MQ Advanced 功能。

SSLClientOutboundSNI

指定当 MQIPT 启动到路由目标的 TLS 连接时 服务器名称指示 (SNI) 扩展的值。 IBM MQ 队列管理器使用 SNI 在 TLS 握手期间提供正确的证书，或者根据配置将连接路由到目标。

此属性仅适用于使用 SSLClient=true 定义的路由，不能对使用 HTTP=true 定义的路由指定此属性。 如果更改此属性的值，并且 **SSLClient** 设置为 true，那么发出刷新命令时将停止并重新启动路由。

注意: 如果使用通道对象 **CERTLABL** 字段上的证书标签配置了目标通道，那么必须将 **CERTLABL** 设置设置为通道值。 如果在没有通道 SNI 设置的情况下转发客户机，那么将拒绝该客户机，并在远程队列管理器错误日志中显示 MQRC_SSL_INITIALIZATION_ERROR 返回码和 AMQ9673 消息。

该属性的值可以是下列其中一个值:

主机名

SNI 设置为路由目标的主机名。 如果路由连接到使用 SNI 路由请求的负载均衡器或路由器，请使用此选项。 例如， Red Hat® OpenShift® Container Platform Router 使用 SNI 将请求路由到 IBM MQ 队列管理器。

如果路由目标是队列管理器，那么连接请求将在 TLS 握手期间接收远程队列管理器的缺省证书，因此无法使用每个通道的证书。

如果使用 IP 地址指定路由目标，并且无法执行反向 DNS 查找，那么 SNI 为空白。

这是缺省值。

通道

SNI 设置为 IBM MQ 通道名称。 使用此选项以允许每个通道的证书由目标队列管理器使用，如果路由接收到的连接由于下列其中一个原因未包含 SNI 中的通道名称:

- 该路由配置为接受未使用 TLS 通过 SSLServer=false 或 SSLPlainConnections=true 进行保护的连接。
- 连接到路由的应用程序无法设置 SNI，或者配置为将 SNI 设置为 IBM MQ 通道名称以外的值。

中继

如果使用 SSLServer=true 定义了路由，那么出站连接上的 SNI 将设置为该路由的入站连接上接收到的 SNI 的值。 如果路由未配置为接受 TLS 连接，那么 SNI 将设置为目标主机名。

定制

SNI 设置为 **SSLClientCustomOutboundSNI** 属性中指定的值。 如果未指定 **SSLClientCustomOutboundSNI** 属性，那么将设置 SNI，就像使用 SSLClientOutboundSNI=hostname 配置路由一样。

none

未设置 SNI。

SSLClientProtocols

用于限制启用的安全套接字协议集，当 **SSLClient** 设置为 true 时，该安全套接字协议用于向路由目标进行出站连接。

您可以指定多个值，并使用逗号分隔。 如果未指定此属性，那么缺省情况下将启用 TLS 1.2 和 TLS 1.3 。 要启用 TLS 1.2 或 TLS 1.3 以外的协议，必须指定要在此属性中启用的协议，还必须遵循 启用不推荐的协议和 CipherSuites 中的过程在 Java runtime environment 中添加对协议的支持。 您可以指定以下一个或多个值。

表 *118: SSL/TLS* 协议的允许值

值	**协议**
SSLv3	SSL 3.0
TLSv1	TLS 1.0
TLSv1.1	TLS 1.1
TLSv1.2	TLS 1.2

表 *118: SSL/TLS* 协议的允许值 *(*继续*)*	
值	**协议**
TLSv1.3	TLS 1.3

在路由属性中使用**值**列中列出的条目。 **协议**列中的相应条目仅供参考。

SSLClientSiteDN_C
使用此属性来指定国家或地区名称，以选择要发送到 SSL/TLS 服务器的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何国家或地区名称的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientSiteDN_CN
使用此属性来指定公共名称，以选择要发送到 SSL/TLS 服务器的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何通用名称的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientSiteDN_DC
使用此属性来指定域组件名称，以选择要发送到 SSL/TLS 服务器的证书。 证书匹配不区分大小写。 您可以指定多个 DC，并使用逗号分隔。 每个 DC 表示域名中的一个元素，例如，使用逗号分隔多个值时域名 `example.ibm.com` 表示为 `example,ibm,com`。 如果您不指定此属性，那么将接受具有任何域组件名称的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientSiteDN_DNQ
使用此属性来指定域限定符，以选择要发送到 SSL/TLS 服务器的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何域限定符的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientSiteDN_L
使用此属性来指定位置名，以选择要发送到 SSL/TLS 服务器的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何位置名称的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientSiteDN_O
使用此属性来指定组织名称，以选择要发送到 SSL/TLS 服务器的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何组织名称的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientSiteDN_OU
使用此属性来指定组织单元 (OU) 名称，以选择要发送到 SSL/TLS 服务器的证书。 您可以指定多个 OU，并使用逗号分隔。 （在文字逗号前加反斜杠 (\) 字符来匹配文字逗号）。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何 OU 名称的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 此路由的所有连接停止。

SSLClientSiteDN_PC
使用此属性来指定邮政编码，以选择要发送到 SSL/TLS 服务器的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何邮政编码的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientSiteDN_ST
使用此属性来指定状态名称，以选择要发送到 SSL/TLS 服务器的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何州/省名称的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientSiteDN_Street
使用此属性来指定街道名称，以选择要发送到 SSL/TLS 服务器的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何街道名称的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientSiteDN_T
使用此属性来指定标题，以选择要发送到 SSL/TLS 服务器的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何标题的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientSiteDN_UID
使用此属性来指定用户标识，以选择要发送到 SSL/TLS 服务器的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何用户标识的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLClientSiteLabel
使用此属性来指定标签名称，以选择要发送到 SSL/TLS 服务器的证书。 如果您不指定此属性，那么将接受具有任何标签名称的证书。 如果更改此属性（并且 **SSLClient** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLExitData
使用此属性来提供要传递给出口的用户定义的字符串。

SSLExitName
使用此属性来定义当路由充当 SSL/TLS 客户机或 SSL/TLS 服务器时将调用的出口的类名。 名称必须包含任何程序包名，例如 `com.ibm.mq.ipt.exit.TestExit`。

SSLExitPath
使用此属性定义出口的位置，用于装入出口的副本。 名称必须是用于找到类文件的标准名称，或者包含类文件的 `.jar` 文件的名称；例如， `C:\mqipt\exits` 或 `C:\mqipt\exits\exits.jar`。

SSLExitTimeout
使用此属性定义终止连接请求之前 MQIPT 等待出口完成的时长。 值为 `0` 意味着 MQIPT 无限期等待。

SSLPlainConnections
使用此属性来指定到配置为接受入站 SSL/TLS 连接的路由的 MQIPT 侦听器端口的连接是否必需 SSL/TLS。 此属性适用于将 **SSLServer** 或 **SSLProxyMode** 属性设置为 true 的路由。 如果启用，此属性允许连接到路由侦听器端口的未加密连接，这意味着无论连接是否加密，MQIPT 可将所有 IBM MQ 连接转发到队列管理器的侦听器端口。 如果未设置此参数，或者将其设置为 `false`，那么仅允许入站 SSL/TLS 连接。 如果更改此属性，那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLProxyMode
将此属性设置为 `true` 以使路由仅接受 SSL/TLS 客户机连接请求，并将请求直接隧道到目标。 如果更改此属性，那么路由停止并在发出刷新命令时重新启动。 此路由的所有连接停止。 此属性不能与以下属性一起使用：

- **SocksClient**
- **SocksServer**
- **SSLClient**
- **SSLServer**

SSLServer
将此属性设置为 `true` 以使路由充当 SSL/TLS 服务器并接受入局 SSL/TLS 连接。 将 **SSLServer** 设置为 `true` 意味着调用者是另一个充当 SSL/TLS 客户机的 MQIPT，或者是启用了 SSL/TLS 的 IBM MQ 客户机或队列管理器。

如果将 **SSLServer** 设置为 `true`，那么必须使用 **SSLServerKeyRing** 属性指定 SSL/TLS 服务器密钥环，或者通过设置 **SSLServerKeyRingUseCryptoHardware** 属性将 MQIPT 配置为使用加密硬件。

如果更改此属性，那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

此属性不能与以下属性一起使用：

- **SocksServer**
- **SSLProxyMode**

SSLServerCAKeyRing
包含 CA 证书的密钥环文件的标准文件名，用于认证来自 SSL/TLS 客户机的证书。 在 Windows 平台，您必须使用双反斜杠 (\\) 作为文件分隔符。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 此路由的所有连接停止。

SSLServerCAKeyRingPW
用于打开使用 **SSLServerCAKeyRing** 属性指定的 SSL/TLS 服务器 CA 密钥环文件的密码，或者用于在 **SSLServerCAKeyRingUseCryptoHardware** 属性设置为 `true` 时连接到加密硬件密钥库的密码。

该值可以是已使用 **mqiptPW** 命令加密的密码，也可以是包含加密密码的文件的标准文件名。 如果在 Windows 平台上指定文件名，那么必须使用双反斜杠 (\\) 作为文件分隔符。 建议您迁移当前存储在文件中的任何密钥环密码，以使用最新且最安全的保护方法，方法是使用 **mqiptPW** 实用程序对密码进行重新加密。 有关在 MQIPT 配置中加密密码的更多信息，请参阅 加密存储的密码。

如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerCAKeyRingUseCryptoHardware

指定是否将支持 PKCS #11 接口的加密硬件用作用于认证来自 SSL/TLS 客户机的证书的 CA 证书的密钥库。 如果此属性设置为 `true`，那么不能在同一路径上设置 **SSLServerCAKeyRing** 。

如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

将加密硬件与 MQIPT 配合使用是 IBM MQ Advanced 功能。 要使用此功能，还需要使用 MQIPT 路由连接的本地队列管理器具有 IBM MQ Advanced， IBM MQ Appliance， IBM MQ Advanced for z/OS 或 IBM MQ Advanced for z/OS VUE 权利。 当此属性设置为 `true` 时，路由将不会启动，除非 **EnableAdvancedCapabilities** 全局属性设置为确认可以使用 IBM MQ Advanced 功能。

SSLServerAskClientAuth
使用此属性可请求 SSL/TLS 服务器进行 SSL/TLS 客户机认证。 SSL/TLS 客户机必须具有自己的证书才能发送到 SSL/TLS 服务器。 将从密钥环文件中检索证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 此路由的所有连接停止。

SSLServerCipherSuites
要在 SSL/TLS 服务器端使用的 SSL/TLS CipherSuite 的名称。 可以是所支持的一个或多个 CipherSuite。 如果将此选项留空，那么将使用与密钥环中的服务器证书兼容的已启用协议的任何 CipherSuite 。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 此路由的所有连接停止。

SSLServerDN_C
使用此属性可接受从此国家或地区名称的 SSL/TLS 客户机接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何公司名称的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerDN_CN
使用此属性可接受从此公共名称的 SSL/TLS 客户机接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何通用名称的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerDN_DC
使用此属性可接受从此域组件名称的 SSL/TLS 客户机接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 您可以指定多个 DC，并使用逗号分隔。 每个 DC 表示域名中的一个元素，例如，使用逗号分隔多个值时域名 `example.ibm.com` 表示为 `example,ibm,com`。 如果您不指定此属性，那么将接受具有任何域组件名称的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerDN_DNQ
使用此属性可接受从此域限定符的 SSL/TLS 客户机接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何域限定符的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerDN_L
使用此属性可接受从此位置的 SSL/TLS 客户机接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何位置的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerDN_O
使用此属性可接受从此组织的 SSL/TLS 客户机接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何组织的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerDN_OU
使用此属性可接受从此组织单元 (OU) 的 SSL/TLS 客户机接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 您可以指定多个 OU，并使用逗号分隔。 （在文字逗号前加反斜杠 (\) 字符来匹配文字逗号）。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何 OU 名称的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 此路由的所有连接停止。

SSLServerDN_PC
使用此属性可接受从此邮政编码的 SSL/TLS 客户机接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何邮政编码的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerDN_ST
使用此属性可接受从此状态的 SSL/TLS 客户机接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何/省的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerDN_Street
使用此属性可接受从此街道名称的 SSL/TLS 客户机接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何街道名称的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerDN_T
使用此属性可接受从此标题的 SSL/TLS 客户机接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何标题的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerDN_UID
使用此属性可接受从此用户标识的 SSL/TLS 客户机接收的证书。 可以使用星号 (*) 作为名称的前缀或后缀来扩展范围。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何用户标识的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerExit
使用此属性可在路由充当 SSL/TLS 服务器时启用或禁用出口。 这允许您在配置文件中定义出口详细信息，而不必真正使用它们。

SSLServerKeyRing
包含服务器证书的密钥环文件的标准文件名。 在 Windows 平台，您必须使用双反斜杠 (\\) 作为文件分隔符。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerKeyRingPW
用于打开通过 **SSLServerKeyRing** 属性指定的 SSL/TLS 服务器密钥环文件或连接到加密硬件密钥库 (如果 **SSLServerKeyRingUseCryptoHardware** 属性设置为 `true`) 的密码。

该值可以是已使用 **mqiptPW** 命令加密的密码，也可以是包含加密密码的文件的标准文件名。 如果在 Windows 平台上指定文件名，那么必须使用双反斜杠 (\\) 作为文件分隔符。 建议您迁移当前存储在文件中的任何密钥环密码，以使用最新且最安全的保护方法，方法是使用 **mqiptPW** 实用程序对密码进行重新加密。 有关在 MQIPT 配置中加密密码的更多信息，请参阅 加密存储的密码。

如果将 **SSLServer** 设置为 `true`，您必须指定 **SSLServerKeyRingPW**。

如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerKeyRingUseCryptoHardware

指定当 MQIPT 充当 SSL/TLS 服务器时，是否使用支持 PKCS #11 接口的加密硬件作为服务器证书的密钥库。 如果此属性设置为 `true`，那么不能在同一路径上设置 **SSLServerKeyRing** 。

如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

将加密硬件与 MQIPT 配合使用是 IBM MQ Advanced 功能。 要使用此功能，还需要使用 MQIPT 路由连接的本地队列管理器具有 IBM MQ Advanced， IBM MQ Appliance， IBM MQ Advanced for z/OS 或 IBM MQ Advanced for z/OS VUE 权利。 当此属性设置为 `true` 时，路由将不会启动，除非 **EnableAdvancedCapabilities** 全局属性设置为确认可以使用 IBM MQ Advanced 功能。

SSLServerProtocols

用于限制启用的安全套接字协议集，当 **SSLServer** 设置为 `true` 时，该安全套接字协议用于接受路由的路由侦听器端口入站连接。

您可以指定多个值，并使用逗号分隔。 如果未指定此属性，那么缺省情况下将启用 TLS 1.2 和 TLS 1.3 。 要启用 TLS 1.2 或 TLS 1.3 以外的协议，必须指定要在此属性中启用的协议，还必须遵循 启用不推荐的协议和 CipherSuites 中的过程在 Java runtime environment 中添加对协议的支持。 您可以指定以下一个或多个值。

表 *119: SSL/TLS* 协议的允许值

值	**协议**
SSLv3	SSL 3.0
TLSv1	TLS 1.0
TLSv1.1	TLS 1.1
TLSv1.2	TLS 1.2
TLSv1.3	TLS 1.3

在路由属性中使用**值**列中列出的条目。 **协议**列中的相应条目仅供参考。

SSLServerSiteDN_C

使用此属性可指定国家或地区名称，以选择要发送到 SSL/TLS 客户机的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何国家或地区名称的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerSiteDN_CN

使用此属性来指定公共名称，以选择要发送到 SSL/TLS 客户机的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何通用名称的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerSiteDN_DC

使用此属性来指定域组件名称，以选择要发送到 SSL/TLS 客户机的证书。 证书匹配不区分大小写。 您可以指定多个 DC，并使用逗号分隔。 每个 DC 表示域名中的一个元素，例如，使用逗号分隔多个值时域名 `example.ibm.com` 表示为 `example,ibm,com`。 如果您不指定此属性，那么将接受具有任何域组件名称的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerSiteDN_DNQ

使用此属性来指定域限定符，以选择要发送到 SSL/TLS 客户机的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何域限定符的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerSiteDN_L

使用此属性来指定位置名，以选择要发送到 SSL/TLS 客户机的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何位置名称的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerSiteDN_O
使用此属性来指定组织名称，以选择要发送到 SSL/TLS 客户机的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何组织名称的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerSiteDN_OU
使用此属性来指定组织单元 (OU) 名称，以选择要发送到 SSL/TLS 客户机的证书。 您可以指定多个 OU，并使用逗号分隔。 （在文字逗号前加反斜杠 (\) 字符来匹配文字逗号）。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何 OU 名称的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 此路由的所有连接停止。

SSLServerSiteDN_PC
使用此属性来指定邮政编码，以选择要发送到 SSL/TLS 客户机的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何邮政编码的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerSiteDN_ST
使用此属性来指定状态名称，以选择要发送到 SSL/TLS 客户机的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何州/省名称的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerSiteDN_Street
使用此属性来指定街道名称，以选择要发送到 SSL/TLS 客户机的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何街道名称的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerSiteDN_T
使用此属性来指定标题，以选择要发送到 SSL/TLS 客户机的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何标题的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerSiteDN_UID
使用此属性来指定用户标识，以选择要发送到 SSL/TLS 客户机的证书。 证书匹配不区分大小写。 如果您不指定此属性，那么将接受具有任何用户标识的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

SSLServerSiteLabel
使用此属性来指定标签名称，以选择要发送到 SSL/TLS 客户机的证书。 如果您不指定此属性，那么将接受具有任何标签名称的证书。 如果更改此属性（并且 **SSLServer** 设置为 `true`），那么路由停止并在发出刷新命令时重新启动。 路由的所有连接停止。

StoredCredentialsFormat
使用此属性来指示密码属性的值是否使用 MQIPT 从 IBM MQ 9.1.5 中支持的加密密码格式。 MQIPT 几乎始终可以检测是否以加密密码格式指定了密码。 仅需要在 MQIPT 无法自动区分加密密码和纯文本密码或文件名的不太可能的场景中设置此属性。

该值可以是下列任一值：

已加密
密码属性包含格式为 MQIPT 从 IBM MQ 9.1.5 支持的加密密码。

COMPAT
密码属性包含纯文本密码，或者对于密钥环密码，包含加密密码的文件的名称。

TCPKeepAlive
将此属性设置为 `true` 以启用 TCP/IP 保持活动包的定期发送，防止此路由上的连接变得空闲。 这样减少防火墙或路由器中断 MQIPT 连接的机会。 TCP/IP 保持活动包的发送由操作系统的调整参数控制；有关如何调整保持活动的更多详细信息，请查看操作系统文档。 如果不设置此参数，或将其设置为 `false`，将不发送保持活动包。

跟踪
此路由所需要的跟踪级别。 启用一个路由的跟踪不会启用任何其他路由的跟踪。 如果您需要跟踪多个路由，您必须将 **Trace** 属性添加到要跟踪的每个路由的 `[route]` 部分。

此属性的值可以是下列其中一项：

0
未启用跟踪

任何正整数
已启用跟踪

缺省值为 0。

如果 [route] 部分不包含 **Trace** 属性，那么将使用 [global] 部分中的 **Trace** 属性。 有关与路由不相关的跟踪线程的信息，请参阅 [global] 部分的 **Trace**。 如果此属性的更改影响路由，那么发出刷新命令时使用新值。 所有连接立即使用新值。 路由不停止。

TraceUserData
在为此路由启用跟踪时，此路由所跟踪的网络传输中接收和发送的用户数据量。 该值可以是下列任一值：

0
未跟踪任何用户数据。

全部
跟踪所有用户数据。

numberOfBytes
跟踪指定的数据字节数，包括传输段头 (TSH)。 指定的值必须大于 15。

UriName

使用 HTTP 代理时，可以使用此属性更改资源的统一资源标识的名称，尽管大多数配置中使用缺省值就足够了。

```
HTTP://destination:destination_port/mqipt
```

如果更改此属性（并且 **HTTP** 设置为 true），那么路由停止并在发出刷新命令时重新启动。

mqiptAdmin 属性

mqiptAdmin 命令从启动命令时指定的属性文件中读取配置属性。

可以在 **mqiptAdmin** 命令使用的属性文件中指定以下属性。 属性名是区分大小写的。

PasswordProtectionKeyFile
包含用于对 **SSLClientCAKeyRingPW** 属性中指定的信任库密码进行加密的加密密钥的文件的名称。 如果未指定此属性，那么将使用缺省加密密钥来解密密码。 用于加密 **mqiptAdmin** 信任库密码的加密密钥可以与用于加密 mqipt.conf 配置文件中的密码的加密密钥不同。

SSLClientCAKeyRing
要用于连接到 MQIPT TLS 命令端口的 PKCS#12 信任库的文件名。 信任库应包含对 MQIPT TLS 命令端口配置为使用的服务器证书进行签名的 CA 的 CA 证书。 必须对文件名中的反斜杠 (\) 字符进行转义，并将其指定为双反斜杠 (\\)。

SSLClientCAKeyRingPW
用于访问使用 **SSLClientCAKeyRing** 属性指定的信任库的加密密码。 必须使用 **mqiptPW** 命令对密码进行加密，并将此属性的值设置为 **mqiptPW** 输出的字符串。

声明

本信息是为在美国国内供应的产品和服务而编写的。

IBM 可能在其他国家或地区不提供本文档中讨论的产品、服务或功能。 有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。 任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。 只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。 但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 可能已拥有或正在申请与本文档内容有关的各项专利。 提供本文档并未授予用户使用这些专利的任何许可。 您可以用书面方式将许可查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区: International Business Machines Corporation “按现状” 提供本出版物，不附有任何种类的 (无论是明示的还是暗含的) 保证，包括但不限于暗含的有关非侵权，适销和适用于某种特定用途的保证。 某些国家或地区在某些交易中不允许免除明示或暗含的保证。 因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。 此处的信息将定期更改；这些更改将编入本资料的新版本中。 IBM 可以随时对本资料中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。 那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation
软件互操作性协调员，部门 49XA
北纬 3605 号公路
罗切斯特，明尼苏达州 55901
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。 因此，在其他操作环境中获得的数据可能会有明显的不同。 有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。 此外，有些测量是通过推算而估计的， 实际结果可能会有差异。 本文档的用户应当验证其特定环境的适用数据。

© Copyright IBM Corp. 2007, 2025

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。 IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。 有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含日常商业运作所使用的数据和报表的示例。 为了尽可能全面地说明这些数据和报表，这些示例包括个人、公司、品牌和产品的名称。 所有这些名字都是虚构的，若现实生活中实际业务企业使用的名字和地址与此相似，纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。 如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。 这些示例并未在所有条件下作全面测试。 因此，IBM 不能担保或默示这些程序的可靠性、可维护性或功能。

如果您正在查看本信息的软拷贝，图片和彩色图例可能无法显示。

编程接口信息

编程接口信息 (如果提供) 旨在帮助您创建用于此程序的应用软件。

本书包含有关允许客户编写程序以获取 IBM MQ 服务的预期编程接口的信息。

但是，该信息还可能包含诊断、修改和调优信息。 提供诊断、修改和调优信息是为了帮助您调试您的应用程序软件。

要点: 请勿将此诊断，修改和调整信息用作编程接口，因为它可能会发生更改。

商标

IBM IBM 徽标 ibm.com 是 IBM Corporation 在全球许多管辖区域的商标。 当前的 IBM 商标列表可从 Web 上的 “Copyright and trademark information”www.ibm.com/legal/copytrade.shtml 获取。 其他产品和服务名称可能是 IBM 或其他公司的商标。

Microsoft 和 Windows 是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

Linux 是 Linus Torvalds 在美国和/或其他国家或地区的商标。

此产品包含由 Eclipse 项目 (https://www.eclipse.org/) 开发的软件。

Java 和所有基于 Java 的商标和徽标是 Oracle 和/或其附属公司的商标或注册商标。

部件号:

(1P) P/N: